

PAULO HIROAQUI RUIZ NAKASHIMA

**ALOCAÇÃO DE GÁS DE ELEVAÇÃO
EM CAMPOS DE PETRÓLEO:
MODELOS E ALGORITMOS**

**FLORIANÓPOLIS
2007**

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**ALOCAÇÃO DE GÁS DE ELEVAÇÃO
EM CAMPOS DE PETRÓLEO:
MODELOS E ALGORITMOS**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção
do grau de Doutor em Engenharia Elétrica.

Paulo Hiroaqui Ruiz Nakashima

Florianópolis, Outubro de 2007.

Alocação de Gás de Elevação em Campos de Petróleo: Modelos e Algoritmos

Paulo Hiroaqui Ruiz Nakashima

‘Esta Tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica, Área de Concentração em *Automação e Sistemas*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Eduardo Camponogara, Ph.D.
Orientador

Katia Campos de Almeida, Ph.D.
Coordenadora do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Eduardo Camponogara, Ph.D.
Presidente

Marcus Vinicius Soledade Poggi de Aragão, Ph.D. PUC-RJ

Mario Cesar Mello Massa de Campos, Dr. CENPES

Julio Elias Normey Rico, Dr. UFSC

Ubirajara Franco Moreno, Dr. UFSC

*Dedico este trabalho a meus pais, Hiroaqui e Ivone, e a minha tia Alice, pela educação e o apoio
que me permitiram chegar até aqui.*

AGRADECIMENTOS

Ao Professor Eduardo Camponogara, pela orientação, amizade, paciência, apoio, e pela confiança depositada em mim para a realização deste trabalho.

Ao jovem camarada Agostinho Plucênio, pela amizade e por sua grande experiência compartilhada com todos, originando valiosas idéias utilizadas aqui.

Aos Professores Augusto, Daniel e Julio, coordenadores do PRH-34 durante o período deste doutorado, pelo total apoio a esta pesquisa.

A todos os meus amigos (sem citar nomes pra não correr o risco de esquecer alguém) que estiveram comigo durante esta minha ótima fase em Florianópolis.

A toda minha família, que soube compreender minha ausência e me apoiou nos momentos difíceis.

A Agência Nacional de Petróleo, que financiou parte desta pesquisa.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

Alocação de Gás de Elevação em Campos de Petróleo Modelos e Algoritmos

Paulo Hiroaqui Ruiz Nakashima

Outubro/2007

Orientador: Prof. Dr. Eduardo Camponogara

Co-Orientador: Prof. Dr. Daniel Pagano

Área de Concentração: Automação e Sistemas

Palavras-chave: Algoritmos, Indústria de Petróleo, Métodos de Otimização, Pesquisa Operacional

Número de Páginas: xiii + 110

O sistema de elevação artificial de petróleo por injeção contínua de gás, conhecido como *continuous gas-lift* ou simplesmente *gas-lift* é um dos mais utilizados no Brasil e no mundo, devido ao seu baixo custo relativo e eficiência satisfatória em uma vasta gama de condições de operação. Neste sistema de produção, a alocação da taxa disponível de injeção de gás para um grupo de poços é um problema de grande relevância. A decisão da taxa de injeção de gás para cada poço normalmente não é ótima, pois é baseada em regras *ad hoc*. Sendo assim, ganhos substanciais podem ser alcançados se estas decisões forem substituídas por soluções ótimas.

Neste trabalho, apresentamos algumas abordagens para a solução do problema de maximizar o lucro obtido com a venda de hidrocarbonetos extraídos de um campo com poços operando via *gas-lift*. As soluções desenvolvidas utilizam Programação Dinâmica, Programação Linear Inteira Mista aliada à Teoria de Desigualdades Válidas e algoritmos *Branch-and-Bound* e *Branch-and-Cut* e fornecem a taxa de injeção de gás ótima para cada poço satisfazendo restrições como por exemplo a taxa de injeção disponível e os limites de processamento de fluidos produzidos.

As abordagens desenvolvidas se mostram eficientes para a aplicação em questão. As soluções encontradas possuem alta qualidade (próximas das soluções ótimas), e o custo computacional para a obtenção das mesmas é baixo. As técnicas de solução propostas aqui constituem excelentes ferramentas de suporte à decisão, podendo substituir as decisões sub-ótimas normalmente fornecidas por pacotes de otimização comerciais.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Lift-Gas Allocation for Oil Fields Models and Algorithms

Paulo Hiroaqui Ruiz Nakashima

October/2007

Advisor: Prof. Dr. Eduardo Camponogara

Co-Advisor: Prof. Dr. Daniel Pagano

Research Area: Systems and Automation

Keywords: Algorithms, Oil Industry, Optimization Methods, Operational Research

Pages: xiii + 110

The artificial lifting method known as continuous gas-lift, or gas-lift for short, is one of the most used in Brazil and in the world, due to its relatively low cost and good efficiency on a wide range of operational conditions. For this production system, the available lift-gas injection rate allocation for a cluster of wells is an important problem. The injection rate for each well is normally below the optimal point, because it is based on ad-hoc rules. Thus, substantial improvements may be reached if these decisions were substituted by the optimal solution.

In this work, we show some approaches to solve the problem of maximize the profit obtained with hydrocarbons extracted from an oilfield with wells producing via gas-lift. The developed solutions use Dynamic Programming, Mixed Integer Linear Programming allied to Valid Inequalities Theory, Branch-and-Bound and Branch-and-Cut algorithms, providing the optimal injection rate for each well, satisfying constraints such as the available lift-gas injection rate and produced fluid handling capacities.

The developed approaches are very efficient for this particular application. The solutions shown here have high quality (close to the optimal ones), and the computational cost to obtain them is low. The solution techniques proposed here are excellent tools to support decisions, and can substitute the sub-optimal solutions normally provided by commercial optimization software.

Sumário

1	Introdução	1
1.1	O Método de <i>Gas-Lift</i>	2
1.2	Revisão Bibliográfica	4
1.3	Objetivos	6
1.4	Organização do Documento	8
2	Fundamentos	10
2.1	Otimização	10
2.2	Programação Dinâmica	11
2.2.1	Programação Dinâmica para o Problema da Mochila	12
2.3	Linearização por Partes	14
2.3.1	Modelo com pesos por ponto (Modelo I)	15
2.3.2	Modelo com pesos sequencial (Modelo II)	16
2.3.3	Modelo de Sherali (Modelo III)	18
2.4	Definições de Poliedro e Dimensão	19
2.5	Descrição de Poliedros através de Facetas	20
2.5.1	Caracterização de Facetas	21
2.6	Desigualdades Válidas para o Problema da Mochila	22
2.7	Sumário	24

3	Solução de $P(\emptyset)$ Utilizando Programação Dinâmica	25
3.1	Formulação do Problema	25
3.2	Solução de $P(\emptyset)$	28
3.2.1	Recorrências	29
3.2.2	Algoritmo PD para $P(\emptyset)$	29
3.3	Resultados Numéricos	31
3.3.1	Comparações com Resultados Publicados	33
3.4	Generalizações do Problema	36
3.4.1	Solução de $P_W(\emptyset)$	37
3.4.2	Incertezas nas $WPCs$	38
3.5	Desigualdades válidas para $P(\emptyset)$	40
3.6	Sumário	42
4	Solução Recursiva de $P(G)$	44
4.1	Formulação do Problema	44
4.2	Solução de $P(F)$	46
4.2.1	Recursões	47
4.2.2	Algoritmo PD para $P(F)$	48
4.2.3	Resultados Numéricos	51
4.3	Solução de $P(G)$	52
4.3.1	Recursões	53
4.3.2	Algoritmo Recursivo para $P(G)$	54
4.3.3	Resultados Numéricos	57
4.4	Extensões	58
4.5	Sumário	59

5	Solução de $P(\emptyset)$ Utilizando Programação Linear Inteira Mista	60
5.1	Linearização por Partes Aplicada a P	60
5.2	Desigualdades Válidas	63
5.3	<i>Lifting</i> de Coberturas Mínimas	73
5.4	Sobre os Fatores de <i>Lifting</i>	76
5.5	Resultados Numéricos	78
5.6	Sumário	80
6	Solução de $P^5(\emptyset)$ Utilizando Programação Linear Inteira Mista	81
6.1	Formulação do Problema	81
6.2	Desigualdades Válidas para $P^5(\emptyset)$	88
6.3	Resultados Numéricos	94
6.4	Sumário	96
7	Conclusões	97
7.1	Pesquisa Futura	98
A	Ajuste de Curvas	100
B	Desigualdades Válidas para $P(G)$	102

Lista de Figuras

1.1	Elementos que compõe um conjunto de poços de petróleo que produzem via <i>gas-lift</i> .	3
1.2	Esquema de poço operado via injeção contínua de gás [14].	4
1.3	Típica curva de desempenho de elevação para um poço operado via injeção contínua de gás.	5
1.4	Curvas de desempenho de <i>gas-lift</i> para poços dos tipos A, B, C e D.	6
2.1	Linearização por partes para o modelo por pesos	16
2.2	Linearização por partes para o modelo sequencial	17
2.3	Linearização por partes para o modelo de Sherali	18
4.1	Ilustração do algoritmo PD para $P(F)$. O grafo da esquerda define as restrições de precedência de ativação dos poços. Para uma ordem topológica $T = \langle 1, 2, 3, 4, 5, 6 \rangle$, os conjuntos de sub-problemas resolvidos e as tabelas geradas pelo algoritmo são mostrados na árvore da direita, que indica a ordem na qual os problemas são resolvidos. Por exemplo, $5 : \mathcal{P}_{\{5,6\}} \rightarrow \mathcal{J}_{\{5,6\}}$ significa que o conjunto de sub-problemas $\mathcal{P}_{\{5,6\}}^y$ foi o quinto a ser resolvido para obter $\mathcal{J}_{\{5,6\}}^y$, para $y \in \{0, 1\}$	50
4.2	Ilustração dos algoritmos recursivos que resolvem $P(G)$. O grafo de restrições de precedência G possui 5 nós como ilustrado na raiz da árvore de sub-problemas. A ordem topológica de G usada para guiar a solução é $T = \langle 3, 1, 2, 4, 5 \rangle$. Cada nó indica um conjunto de sub-problemas resolvido pelo algoritmo: $\mathcal{P}(1, 2, 4)$ representa o conjunto de sub-problemas $\mathcal{P}(G[\{1, 2, 4\}])$	56
5.1	Ilustração da linearização por partes da função do fluxo de saída de um poço n utilizando o modelo de Sherali (Seção 2.3.3).	62

6.1	Ilustração da linearização por partes da função do fluxo de saída de um poço n utilizando o modelo de pesos por ponto (Seção 2.3.1).	85
-----	--	----

Lista de Tabelas

2.1	Valores de $f_k(\lambda)$ e de $p_k(\lambda)$ para uma instância do problema da mochila	14
3.1	Parâmetros de uma instância de $P(\emptyset)$ com 6 poços.	30
3.2	Tabela $J_{m,n}$, com destaque para a solução ótima $J_{M,1} = J_{10,1}$	31
3.3	Tabela auxiliar $I_{m,n}$, com destaque para a alocação ótima de gás $q_i = (7,4251; 7,6954; 7,4406; 4,0000; 4,0000; 7,$	
3.4	Qualidade média da solução e tempo médio de execução do algoritmo com relação ao nível de discretização M	32
3.5	Resultados da alocação de gás de elevação, <i>Economic Slope</i> [18] e PD	33
3.6	Resultados da alocação de gás de elevação, <i>Equal Slope, Ex-In</i> [6] e PD	34
3.7	Resultados da alocação de gás de elevação, <i>Separable Programming</i> [40] e PD	34
3.8	Alocação de gás de elevação, cenário sem taxas de partida: comparação entre SQP [1] e PD	35
3.9	Alocação de gás de elevação, cenário com taxas de partida: comparação entre SQP [1] e PD	35
3.10	Resultados de alocação de gás de elevação: comparação entre GOAL [36] e PD	36
3.11	Comparação entre desempenho do pior caso e caso médio	39
4.1	Soluções de $P(F)$ para uma instância de 6 poços, com q_i^{max} variável e $M = 200$	51
4.2	Qualidade de aproximação e tempo de execução para $P_M(F)$	52
4.3	Qualidade de aproximação e tempo de execução para $P_M(G)$	58
5.1	Níveis de injeção para uma instância de 6 poços, $q_i^{max} = 6,5$	64

5.2	Limites inferiores para os fatores de <i>lifting</i> , $\pi_{n,k}$.	72
5.3	Fatores de <i>lifting</i> exatos, $\beta_{n,k}$.	74
5.4	Limites superiores para os fatores de <i>lifting</i> , $\beta'_{n,k}$.	76
5.5	Experimentos com ILOG CPLEX	79
5.6	Experimentos com <i>GNU MILP solver</i>	80
6.1	Poço 1	89
6.2	Poço 2	89
6.3	Poço 3	89
6.4	Impacto das coberturas- y na velocidade de solução para instâncias com 32 poços.	95
6.5	Impacto das coberturas- y na velocidade de solução para instâncias com 64 poços.	95

Capítulo 1

Introdução

A produção de hidrocarbonetos é realizada através da perfuração de poços em locais que apresentam formações geológicas ricas em óleo e gás, chamadas de reservatórios. Quando a pressão de um reservatório é suficientemente elevada para superar a pressão hidrostática e a perda de carga na coluna de produção, somados à perda de carga nas instalações de superfície, os fluidos nele contidos são capazes de alcançar a superfície naturalmente. Os poços que produzem desta forma são denominados *poços surgentes*. Quando a pressão de um reservatório é relativamente baixa, ou quando a vazão de um poço está muito abaixo da ideal, há a necessidade de uma suplementação da energia natural do reservatório através de *elevação artificial*. Utilizando equipamentos específicos reduz-se a pressão hidrostática no fundo do poço, resultando em um diferencial maior de pressão entre o reservatório e o interior do poço, com um consequente aumento de vazão. A elevação artificial é amplamente utilizada em poços submarinos, especialmente em águas profundas. Os métodos de elevação artificial mais comuns na indústria de petróleo são:

- *gas-lift* contínuo e intermitente;
- bombeio centrífugo submerso;
- bombeio mecânico com hastes; e
- bombeio por cavidades progressivas.

A seleção do melhor método de elevação artificial para um determinado poço ou campo sempre visa a maximização do uso da energia de produção naturalmente disponível em um reservatório. Esta seleção depende de vários fatores, como número de poços, quantidade de areia trazida pelos fluidos produzidos, razão gás-líquido, capacidade de vazão disponível, profundidade do reservatório, viscosidade dos fluidos, disponibilidade de energia, investimento e custo operacional.

A elevação artificial por injeção contínua de gás, mais conhecida como *continuous gas-lift* ou simplesmente *gas-lift*, é uma das formas mais utilizadas de elevação artificial para produção de petróleo [24], [32], devido ao seu amplo campo de aplicação, além de ser o método que mais se assemelha ao processo de fluxo natural. Seu único requisito básico é que exista uma quantidade de gás de elevação disponível que seja economicamente viável. O sistema de *gas-lift* é eficiente em uma vasta gama de condições de operação: é muito versátil em termos de vazão e profundidade, propício para poços que produzem fluidos com alto teor de areia e elevada razão gás-líquido, exige investimentos relativamente baixos para poços profundos, e seus custos de instalação e manutenção são mais baixos que os dos demais métodos de elevação artificial [39]. A seguir, descrevemos os princípios básicos do método de elevação artificial por injeção contínua de gás.

1.1 O Método de *Gas-Lift*

A Figura 1.1 ilustra esquematicamente um conjunto de poços equipados para produzir por *gas-lift*. O sistema é composto por:

- fonte de gás a alta pressão (compressores);
- controlador de injeção de gás na superfície (*choke*);
- controlador de injeção de gás de subsuperfície (válvulas de *gas-lift*, Figura 1.2);
- equipamentos para separação e armazenamento dos fluidos produzidos (separadores e tanques).

A Figura 1.2 mostra o esquema típico de um poço operado via injeção contínua de gás. O sistema de *gas-lift* funciona da seguinte maneira. Gás a alta pressão é injetado no espaço anular entre o revestimento do poço e o tubo de produção, de forma controlada e contínua. O gás entra no tubo de produção através de uma válvula de *gas-lift* em um ponto próximo ao fundo do poço, e mistura-se ao fluido existente na coluna diminuindo sua densidade média. Assim, a componente gravitacional da pressão no fundo do poço diminui, até que esta diminuição cause um diferencial de pressão entre a formação e a face dos canhoneados (*perforated zone*) que seja suficiente para atingir a taxa de produção desejada. O controle da injeção de gás no poço é feito na superfície através de um regulador de fluxo (*choke*).

A Figura 1.3, denominada curva de desempenho de *gas-lift*, ou *WPC (Well Performance Curve)*, ilustra o comportamento da vazão de saída de acordo com a taxa de injeção de gás em um poço que produz via *gas-lift*. Até certo limite, aumentando-se a quantidade de gás na coluna de produção diminui-se o gradiente médio de pressão, com consequente diminuição da pressão de fluxo no fundo e aumento de vazão. Entretanto, a taxa de injeção de gás não pode ser aumentada indefinidamente, pois a altas taxas de injeção a perda de carga no tubo de produção devido ao atrito se torna dominante,

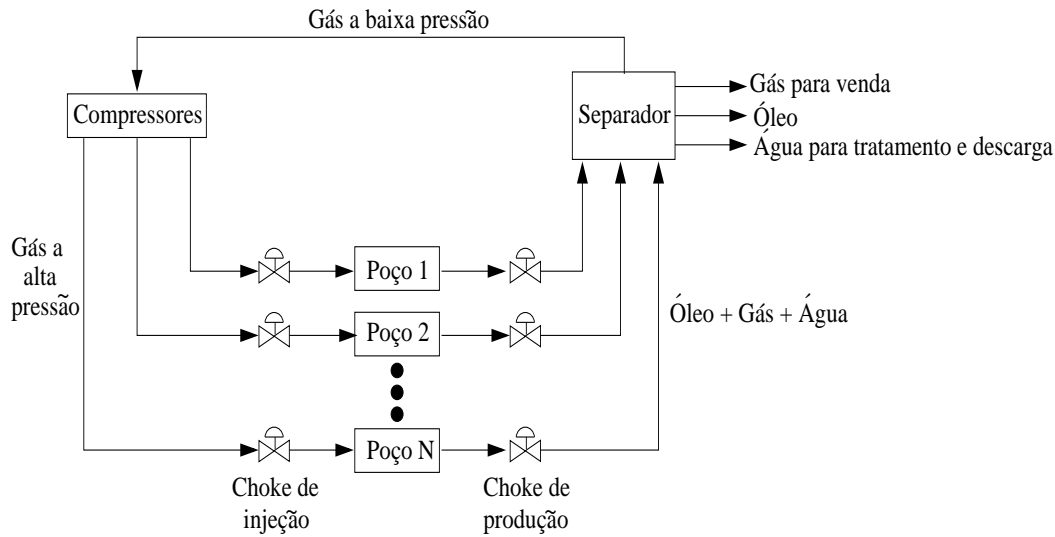


Figura 1.1: Elementos que compõe um conjunto de poços de petróleo que produzem via *gas-lift*.

o que tende a reduzir as taxas de produção, além de poder causar danos como erosão na tubulação. Tipicamente, poços que operam via injeção contínua de gás possuem um comportamento estável a taxas elevadas de injeção e comportamento instável a baixas taxas. As condições instáveis de operação a baixas taxas de injeção fazem com que um poço não produza a máxima quantidade possível de óleo. Operar um poço via injeção contínua de gás sob condições instáveis torna o controle e alocação da produção muito difícil, e os picos de surgência (golfadas) representam condições de operação bastante severas para as instalações de produção. Outra variável importante que deve ser controlada é o diferencial de pressão entre o reservatório e o fundo do poço (*drawdown*). Este diferencial, que está diretamente relacionado à taxa de injeção de gás, pode causar danos à formação através de efeitos indesejáveis como o aumento da produção de areia e elevação do contato água-óleo (conificação), diminuindo a vida útil e o fator de recuperação do reservatório. Portanto, existe um limite mínimo para a taxa de injeção de gás, para que a operação instável do poço seja evitada, e também um limite máximo, além do qual podem ser causados danos à tubulação e ao reservatório.

Existem 4 tipos básicos de *WPC*, como mostra a Figura 1.4 [1]. Os poços do tipo *A* são os chamados surgentes, aqueles que produzem fluidos mesmo sem injeção de gás. Os poços do tipo *B* são aqueles que respondem imediatamente à injeção de gás, ou seja, se $q_p = 0$ para $q_i = 0$ e $q_p > 0$ para $q_i > 0$. Os poços do tipo *C* possuem *taxa de partida* (*kick-off rate*), ou seja, só começam a produzir a partir de uma determinada taxa de injeção de gás. Os poços do tipo *D* também possuem taxa de partida, porém começam a produzir a uma taxa maior que zero.

A produção dos poços é normalmente otimizada com base em restrições técnicas e objetivos econômicos e estratégicos, enquanto leva em conta regras de segurança e de produção, política de exploração do reservatório (máximo escoamento por poço, quotas de produção), cuidados na inter-

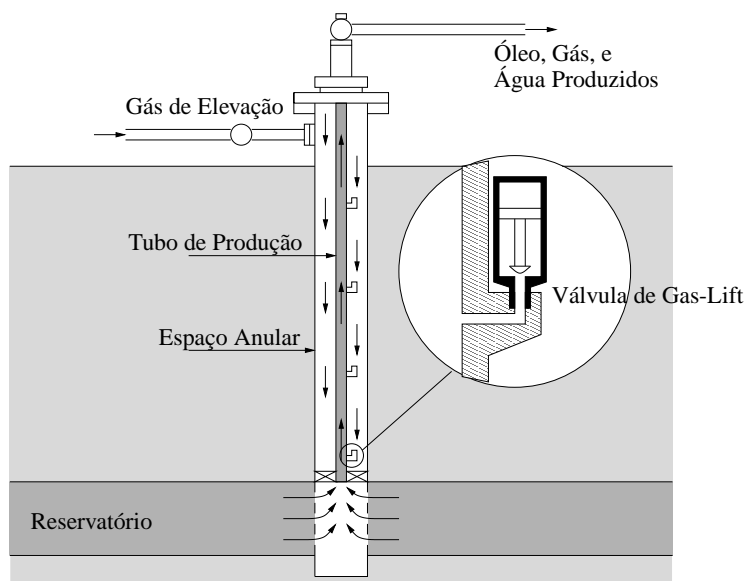


Figura 1.2: Esquema de poço operado via injeção contínua de gás [14].

face entre a parede do poço e a formação (controle de areia) e capacidade das instalações. Neste trabalho, a função objetivo do problema de otimização é definida pelo benefício econômico do uso do gás de elevação (maximizar a quantidade de petróleo recuperado por quantidade de gás injetado), e as restrições do problema são impostas pela taxa de injeção de gás disponível, pelos limites de injeção de gás mínimo (estabilidade da produção) e máximo (segurança da tubulação, preservação do reservatório e aproveitamento do potencial de elevação do gás), e pelos limites impostos pelas instalações de superfície, como capacidade de separação, armazenamento e tratamento dos fluidos produzidos.

Na seção seguinte apresentamos a revisão bibliográfica dos trabalhos relacionados ao problema de alocação de gás de elevação encontrados na literatura, onde são apontadas algumas lacunas que este trabalho pretende preencher.

1.2 Revisão Bibliográfica

O problema de alocação ótima da taxa disponível de injeção de gás de elevação vem sendo estudado desde a década de 70, quando um procedimento analítico *ad hoc* foi apresentado em [35], para determinar a distribuição de gás mais lucrativa entre os poços. Em [18], é apresentado um método de otimização denominado *equal-slope* que se utiliza de uma expressão matemática para encontrar a taxa ótima de injeção de gás, baseando-se na inclinação da curva que relaciona taxa de injeção com o lucro obtido para cada poço, conhecida como curva de desempenho de *gas-lift*. Em essência, este método consiste na solução das equações que caracterizam as condições de primeira ordem (Karush-

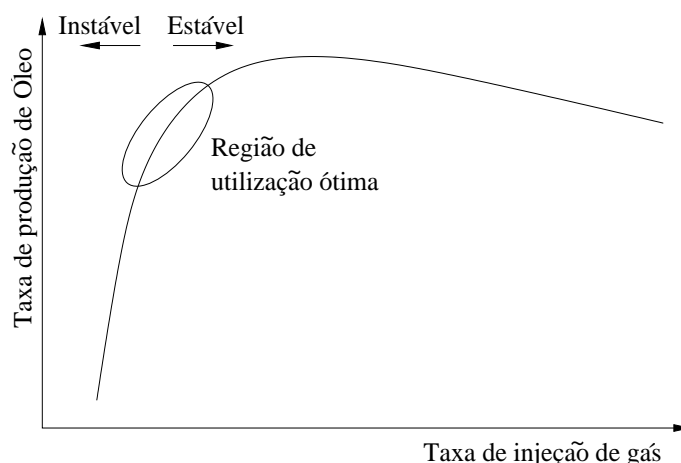


Figura 1.3: Típica curva de desempenho de elevação para um poço operado via injeção contínua de gás.

Kuhn-Tucker) para ótimo local de um problema de maximização côncavo. Apesar de estes métodos fornecerem a solução ótima quando existe gás suficiente para atender todos os poços, eles não podem ser aplicados quando há poços dos tipos *C* e *D* (poços com taxa de partida), pois suas curvas de desempenho são não-côncavas. Além disso, estes métodos não são capazes de tratar outras restrições além da quantidade limitada de gás de elevação.

Abordagens mais flexíveis e formais foram propostas em [32] e [1], baseadas em técnicas não-lineares de otimização quasi-Newton e programação quadrática sequencial (ou *SQP* - *Sequential Quadratic Programming*), respectivamente. Estas abordagens são capazes de lidar com limites nas taxas de injeção de gás e a restrição da quantidade disponível de gás de elevação, mas não admitem a existência de poços dos tipos *C* e *D*.

Métodos heurísticos foram utilizados para resolver o problema da alocação de gás em [23] e [6], sendo que o primeiro baseia-se em um algoritmo genético e o segundo combina uma exploração estocástica do domínio de soluções, aliada a um cálculo heurístico de uma direção de descenso. Métodos heurísticos não são métodos de otimização global, ou seja, as soluções obtidas podem ser ótimos locais. Além disso, o custo computacional para a exploração do espaço de soluções pode ser bastante alto.

O problema de maximização da produção de óleo sob múltiplas restrições nas instalações (*facility constraints*) é considerado em [15] e [40]. Em [15], as curvas de desempenho dos poços são aproximadas por funções lineares por partes (*piecewise linear functions*), e o problema linear resultante é resolvido pelo método Simplex. Porém, para que o método Simplex pudesse ser utilizado, as variáveis inteiras foram consideradas contínuas, e as soluções encontradas podem ser infactíveis (não inteiras), necessitando ser tratadas *a posteriori*. Em [40], variáveis binárias são adicionadas à formulação para

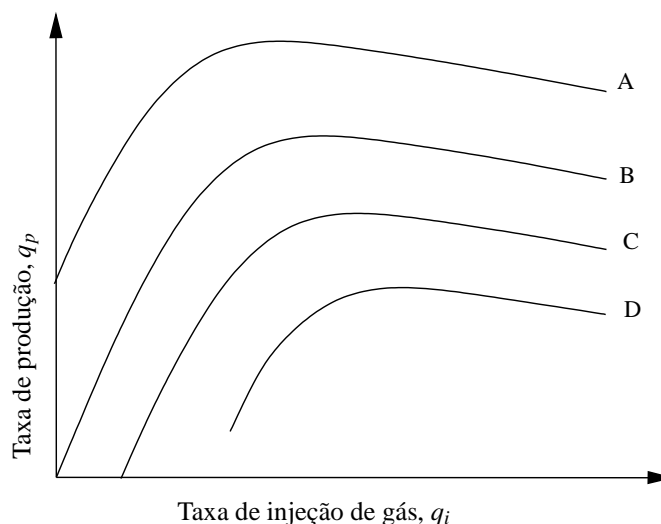


Figura 1.4: Curvas de desempenho de gas-lift para poços dos tipos A, B, C e D.

cada *WPC* não côncava e uma busca *branch-and-bound* é realizada até que a solução ótima seja encontrada. A desvantagem desta abordagem é que algoritmos de *branch-and-bound* puros podem ter tempo de execução e consumo de memória muito altos para instâncias complexas, especialmente em problemas NP-difíceis¹.

Em suma, os modelos existentes não contemplam aspectos importantes do problema, como a existência de taxas de partida, múltiplas restrições nas facilidades de processamento e restrições de precedência de ativação. Além disso, nenhum dos trabalhos apresenta algoritmos eficazes no sentido de tempo computacional e qualidade da solução (aproximar ou encontrar o ótimo global). A seguir, são apresentados os objetivos deste trabalho.

1.3 Objetivos

No sistema de produção de petróleo via *gas-lift*, o problema de alocação de gás de elevação é de bastante relevância. As instalações de compressão de gás representam uma parcela significativa do custo de operação de um campo de petróleo [6], sendo assim é importante que o gás seja utilizado da maneira mais eficiente possível [24]. Normalmente, esta alocação é realizada com base em regras

¹Um problema pertence à classe P (problema polinomial) se pode ser resolvido em tempo polinomial no seu tamanho (e.g., problemas de ordenação, caminhos mínimos em grafos, fluxo máximo em redes, Programação Linear). Um problema pertence à classe NP (problema polinomial não-determinístico) se uma suposta solução para este problema pode ser verificada em tempo polinomial no seu tamanho. Dizemos que um problema é NP-difícil se ele é pelo menos tão difícil quanto qualquer problema em NP. Um problema é NP-completo se qualquer problema em NP pode ser reduzido a este problema em tempo polinomial (e.g., caminho Hamiltoniano, clique máxima em grafos) [38],[42].

ad-hoc [40]. Substituir estas regras por algoritmos de otimização rigorosos pode trazer grandes benefícios. No Brasil, 90% da produção de petróleo provém de poços submarinos, e estima-se que 60% destes poços operam via *gas-lift*. Visto que a necessidade de melhoria do desempenho da tecnologia atual é um fato [2], o aumento, mesmo que pequeno, na eficiência dos processos de produção de petróleo pode resultar em substanciais ganhos econômicos e ambientais.

Os objetivos deste trabalho são a concepção de modelos formais e o projeto, implementação e análise de algoritmos eficientes para resolver o problema de alocação de gás de elevação para um conjunto de poços de petróleo que produzem via *gas-lift*, de modo a maximizar o lucro obtido com a venda de hidrocarbonetos extraídos. Neste documento apresentamos algoritmos de Programação Dinâmica que resolvem este problema, e também formulações de Programação Linear Inteira Mista (ou *MILP* - *Mixed Integer Linear Programming*) para o problema de alocação de gás de elevação sujeito a outras restrições além da taxa de injeção disponível. Para estas formulações são obtidas famílias de desigualdades válidas que são utilizadas para auxiliar a solução através de algoritmos de otimização comerciais e não-comerciais.

Os algoritmos de Programação Dinâmica (doravante chamados algoritmos PD) desenvolvidos neste trabalho possuem as seguintes características, que se tornarão explícitas ao longo do documento:

- resolvem o problema de otimização mesmo quando cada poço possui múltiplas curvas de desempenho de *gas-lift* (*WPCs*), e também sob restrições de precedência de ativação/desativação de poços;
- poços que possuem taxa de partida, ou seja, poços dos tipos *C* e *D* (Figura 1.4), que possuem curvas de desempenho não-côncavas, são tratados com naturalidade;
- uma família de problemas é resolvida, ou seja, problemas para várias taxas disponíveis de injeção de gás, entre 0 e q_i^{max} (taxa máxima de injeção disponível);
- a solução encontrada é muito próxima da solução ótima, e o tempo gasto para o cálculo da solução é pequeno (considerando aplicações para o setor de extração de petróleo); e
- é uma ferramenta de suporte à decisão para o engenheiro de produção de petróleo, de baixo custo e extremamente portátil.

A outra abordagem utilizada neste trabalho é a linearização por partes das *WPCs* e utilização de *MILP* para a obtenção das taxas de injeção ótimas. As vantagens desta abordagem são citadas a seguir, sendo que algumas delas assemelham-se às vantagens mencionadas para o algoritmo PD:

- a transformação do problema não-linear em um problema linear inteiro misto torna mais fácil a introdução e modificação de restrições, além de tornar possível o uso de algoritmos eficientes de Programação Inteira para a busca de uma solução ótima global, ou possíveis soluções de boa qualidade;

- são obtidas famílias de desigualdades válidas que se mostram bastante eficientes na aceleração do tempo de resposta de algoritmos de *Branch-and-Bound* e *Branch-and-Cut*, tornando possível a utilização de *software* de baixo custo para a solução do problema de otimização, como algoritmos de Programação Linear e de *Branch-and-Bound* em plataformas de *software* livre;
- poços com taxa de partida podem ser considerados;
- a solução encontrada é quase-ótima e em tempo razoavelmente baixo;
- seu custo de implementação é baixo e possui grande capacidade de portabilidade e suporte à decisão; e
- os modelos podem ser facilmente estendidos para incorporar outras restrições além das já consideradas.

1.4 Organização do Documento

O Capítulo 2 apresenta o paradigma da Programação Dinâmica, conceitos de funções lineares por partes e de teoria poliedral. O leitor familiarizado com estes tópicos pode dirigir a leitura ao próximo capítulo.

Os capítulos seguintes estão organizados seguindo a cronologia da evolução da pesquisa. No Capítulo 3 é formulada a instância mais básica do problema, denominada $P(\emptyset)$, onde as restrições são a taxa máxima disponível para injeção de gás a ser alocada entre os poços e os limites inferior e superior de taxa de injeção para cada poço. Uma solução baseada em Programação Dinâmica é apresentada, bem como resultados numéricos que comprovam a eficiência dos algoritmos e algumas comparações de desempenho das soluções desenvolvidas neste trabalho com as soluções encontradas na literatura.

O Capítulo 4 contém um algoritmo de Programação Dinâmica e uma solução recursiva, respectivamente para as formulações chamadas de $P(F)$ e $P(G)$, onde além das restrições definidas em $P(\emptyset)$ existem também restrições de precedência de operação entre poços, representando cenários de campos de petróleo onde exista alguma hierarquia de poços produtores. Novamente, resultados numéricos mostram que os algoritmos são bastante eficientes e produzem soluções de boa qualidade.

Neste momento da pesquisa foi percebido que o algoritmo de Programação Dinâmica desenvolvido não era facilmente adaptável quando novas restrições eram incluídas na formulação básica, e desde então passamos a buscar uma solução para $P(\emptyset)$ utilizando Programação Linear Inteira Mista (ou *MILP - Mixed Integer Linear Programming*), mostrada no Capítulo 5. O problema original é transformado em um problema linear por partes e desigualdades válidas são produzidas e incluídas na formulação, de modo a torná-la mais “apertada” e acelerar o processo de solução. A eficiência deste método é comprovada através de experimentos numéricos.

Visto que na abordagem MILP a inclusão de restrições é tratada com naturalidade, passamos a estudar a solução de um problema mais completo, cuja formulação foi denominada $P^5(\emptyset)$, onde são acrescentadas restrições impostas por limites nas instalações de superfície. No Capítulo 6 são apresentados os resultados obtidos. Mais uma vez, são apresentadas famílias de desigualdades válidas que são capazes de acelerar o processo de solução quando introduzidas na formulação, o que é mostrado através de experimentos numéricos.

O Capítulo 7 se refere às considerações finais do trabalho, e algumas direções para pesquisas futuras são apresentadas.

Capítulo 2

Fundamentos

Neste capítulo, apresentamos alguns conceitos básicos do domínio da otimização, bem como o paradigma da Programação Dinâmica, conceitos de funções lineares por partes e de teoria poliedral. O leitor familiarizado com estes tópicos pode proceder diretamente ao Capítulo 3.

2.1 Otimização

A Otimização é a área da Matemática Aplicada que se concentra em calcular valores ótimos para variáveis de decisão de acordo com algum critério de avaliação, ao mesmo tempo que satisfazem restrições de um modelo matemático. A solução de um problema de otimização normalmente possui duas fases: a primeira consiste em transformar o problema em um modelo, e posteriormente implementar um algoritmo capaz de encontrar uma solução adequada para este modelo. A linguagem utilizada para expressar os problemas de maneira declarativa é conhecida como Programação Matemática [41]. Os elementos de um modelo em Programação Matemática são:

Variáveis de decisão : elementos cujos valores definem uma solução para o problema, e.g., quantidades produzidas ou recursos utilizados.

Função objetivo : uma função das variáveis de decisão, que deve ser maximizada ou minimizada, e.g., minimizar custos, reduzir número de homens/hora, maximizar lucros.

Restrições : um conjunto de funções que define o espaço factível de soluções, e.g., limites para recursos, restrições operacionais de um processo de produção, limitações físicas e técnicas.

Um problema geral de otimização pode ser escrito em Programação Matemática como:

$$\begin{array}{ll}\text{Minimize} & f(x) \\ \text{Sujeito a:} & g(x) \leq 0 \\ & h(x) = 0 \\ & x \in \mathbb{R}^n\end{array}$$

onde $f: \mathbb{R}^n \rightarrow \mathbb{R}$ é a função objetivo, $g: \mathbb{R}^n \rightarrow \mathbb{R}^p$ e $h: \mathbb{R}^n \rightarrow \mathbb{R}^q$ são restrições que limitam o espaço de soluções factíveis, e x é o vetor das variáveis de decisão.

Formulado o problema, o próximo passo é encontrar um algoritmo eficiente para o cálculo da solução ótima. Alguns tipos de problema de otimização, como por exemplo os problemas de Programação Linear, possuem algoritmos bastante eficientes para sua solução (método Simplex e método de ponto interior). Porém, existem problemas para os quais não existem algoritmos capazes de encontrar uma solução ótima com a mesma eficiência, como é o caso de alguns problemas de Programação Inteira pertencentes à classe NP-difícil. A seguir apresentamos o paradigma da Programação Dinâmica, bastante utilizado para a criação de algoritmos que resolvem problemas desta classe.

2.2 Programação Dinâmica

Uma das técnicas mais poderosas para a solução de problemas de otimização é quebrá-los em partes menores e mais fáceis de resolver, e posteriormente compor a solução do problema original a partir da solução destes subproblemas. Sempre que é possível quebrar um problema em instâncias menores do mesmo problema, o uso de um algoritmo recursivo se torna aparente. A Programação Dinâmica (PD) é um dos paradigmas de projeto de algoritmos que se baseia em quebrar problemas em problemas menores. Tipicamente remove-se um elemento do problema, resolve-se o problema menor, e então utiliza-se a solução deste problema menor para adicionar o elemento novamente de maneira apropriada.

Algoritmos gulosos, que tomam a melhor decisão local a cada passo, ocasionalmente encontram ótimos globais para certos problemas. Algoritmos de busca exaustiva, que enumeram todas as possibilidades e selecionam a melhor, por definição encontram sempre o ótimo global, porém normalmente a um custo muito alto. A Programação Dinâmica combina as duas técnicas, sistematicamente considerando todas as possíveis decisões e selecionando aquela que prova ser a melhor. Armazenando as consequências de todas as possíveis decisões até o momento, a quantidade de trabalho total é minimizada. A maior limitação da Programação Dinâmica é o número de soluções parciais que devem ser armazenadas. A ordem dos elementos do problema também é importante. Alterar esta ordem significa mudar o problema. Porém, uma vez que esta ordem seja fixada, algoritmos PD são bastante eficientes.

A seguir mostramos um pequeno exemplo, com o objetivo de tornar mais claro o princípio de funcionamento deste tipo de algoritmo. A Programação Dinâmica é uma ferramenta que tem sido aplicada com sucesso a um tipo de problema de otimização chamado de “problema da mochila” (*knapsack problem*) [22]. Diversos problemas de importância estratégica têm uma “mochila” como subproblema, como por exemplo o procedimento de geração de colunas em problemas de corte e empacotamento. Outras vantagens do algoritmo PD serão explicitadas mais adiante.

2.2.1 Programação Dinâmica para o Problema da Mochila

Imagine um alpinista que deve organizar sua mochila com vários itens de seu interesse ao escalar uma montanha, sendo que cada item possui um peso e um valor, e a mochila possui uma capacidade limitada. O desejo do alpinista é maximizar a utilidade (ou valor) dos itens colocados em sua mochila, sem ultrapassar a capacidade de peso suportada por ela. Este problema é conhecido como o problema da mochila (*knapsack problem*, *KP*), e pode ser formulado da seguinte maneira:

$$\begin{aligned}
 KP : \quad z = & \text{Maximizar} \quad \sum_{j=1}^n c_j x_j \\
 \text{Sujeito a:} \quad & \sum_{j=1}^n a_j x_j \leq b \\
 & x_j \in \{0, 1\}, \quad j = 1, \dots, n
 \end{aligned} \tag{2.1}$$

onde x_j assume valor 1 se o item j vai para a mochila, e 0 caso contrário, c_j é o valor do item j , e os coeficientes $\{a_j\}_{j=1}^n$ e b são inteiros positivos representando respectivamente o peso do item j e a capacidade da mochila, com $a_j \leq b$, $j = 1, \dots, n$.

Seja $KP_k(\lambda)$ o subproblema correspondente ao problema original KP restrito aos k primeiros itens ($j = 1, \dots, k$), para uma mochila de capacidade λ :

$$\begin{aligned}
 KP_k(\lambda) : \quad f_k(\lambda) = & \text{Maximizar} \quad \sum_{j=1}^k c_j x_j \\
 \text{Sujeito a:} \quad & \sum_{j=1}^k a_j x_j \leq \lambda \\
 & x_j \in \{0, 1\}, \quad j = 1, \dots, k.
 \end{aligned} \tag{2.2}$$

Resolver o problema original KP significa obter a solução ótima para $KP_n(b)$, ou seja, o valor $z = f_n(b)$.

O problema $KP_k(\lambda)$ possui 2 casos base, de solução trivial:

- $KP_k(0)$ para $k = 1, \dots, n$, pois se a capacidade é zero nenhum item vai para a mochila e, portanto, $f_k(0) = 0$ para $k = 1, \dots, n$;

- $KP_0(\lambda)$ para $\lambda = 0, \dots, b$, pois se temos 0 itens a considerar, nenhum item vai para a mochila, e portanto $f_0(\lambda) = 0$ para $\lambda = 0, \dots, b$.

Consideremos agora a solução do conjunto de subproblemas que consideram apenas o primeiro item, variando a capacidade da mochila (λ) de 0 até b . Claramente, cada um destes subproblemas possui apenas 2 possibilidades de soluções ótimas, $x_1^* = 0$ ou $x_1^* = 1$, ou seja, o item 1 fica fora da mochila ou dentro dela. Considerando cada caso, temos:

- Se $x_1^* = 0$, então concluímos que a solução ótima satisfaz $f_1(\lambda) = f_0(\lambda)$, ou seja, o item 1 não agrega valor à função objetivo e não ocupa espaço na mochila, portanto a solução do problema é a solução correspondente ao problema que considera 0 itens e espaço λ , que já está resolvido;
- Se $x_1^* = 1$, então concluímos que a solução ótima satisfaz $f_1(\lambda) = c_1 + f_0(\lambda - a_1)$, ou seja, colocamos o item 1 na mochila ocupando espaço a_1 e agregando valor c_1 à função objetivo, e somamos o valor ótimo do problema com 0 itens e capacidade $\lambda - a_1$, que já foi resolvido.

Isto nos leva a uma recursão que permite calcular $f_k(\lambda)$ em termos dos subproblemas já resolvidos, ou seja, os valores de $f_s(u)$ com $s < k$ e $u \leq \lambda$. Generalizando o caso acima:

- 1) Se $x_k^* = 0$, então concluímos que a solução ótima satisfaz $f_k(\lambda) = f_{k-1}(\lambda)$;
- 2) Se $x_k^* = 1$, então concluímos que a solução ótima satisfaz $f_k(\lambda) = c_k + f_{k-1}(\lambda - a_k)$, para $a_k \leq \lambda$.

Combinando os casos (1) e (2), obtemos a seguinte recorrência:

$$f_k(\lambda) = \max\{f_{k-1}(\lambda), c_k + f_{k-1}(\lambda - a_k)\} . \quad (2.3)$$

Definindo-se os valores iniciais como $f_0(\lambda) = 0$ para $\lambda \geq 0$ e $f_k(0) = 0$ para $k = 1, \dots, n$, pode-se utilizar a recorrência (2.3) para calcular sucessivamente os valores de f_1, f_2, \dots, f_n para todos os valores inteiros de $\lambda = 1, \dots, b$.

A questão que resta é como encontrar a solução ótima (ou seja, quais itens vão para a mochila) associada ao valor ótimo $f_n(b)$. Podemos manter um indicador $p_k(\lambda)$ que assume valor 0 se $f_k(\lambda) = f_{k-1}(\lambda)$, e valor 1 caso contrário. Portanto para definir qual item vai para a mochila, basta observar os valores de $p_k(\lambda)$ da seguinte maneira:

- se $p_n(b) = 0$, então como $f_n(b) = f_{n-1}(b)$, definimos $x_n^* = 0$ e continuamos o processo com o valor $f_{n-1}(b)$;
- se $p_n(b) = 1$, então $f_n(b) = c_n + f_{n-1}(b - a_n)$, definimos $x_n^* = 1$ e repetimos este procedimento para $f_{n-1}(b - a_n)$;

Tabela 2.1: Valores de $f_k(\lambda)$ e de $p_k(\lambda)$ para uma instância do problema da mochila

	f_0	f_1	f_2	f_3	f_4	p_1	p_2	p_3	p_4
$\lambda = 0$	0	0	0	0	0	0	0	0	0
1	0	0	7	7	7	0	1	0	0
2	0	10	10	10	10	1	0	0	0
3	0	10	17	17	17	1	1	0	0
4	0	10	17	17	17	1	1	0	0
5	0	10	17	17	24	1	1	0	1
6	0	10	10	25	31	1	1	1	1
7	0	10	17	32	34	1	1	1	1

- após n iterações, obteremos a solução ótima.

Calculando o número de operações necessárias para obtermos $z = f_n(b)$, verificamos que o cálculo de $f_k(\lambda)$ para $\lambda = 0, 1, \dots, b$ e $k = 1, \dots, n$ necessita de um número constante de operações. O algoritmo possui tempo de execução $\Theta(nb)$, sendo portanto pseudo-polinomial.

Reproduzimos agora um exemplo numérico extraído de [42], com o objetivo de tornar mais claros os conceitos apresentados nesta seção. Considere a instância de *KP*:

$$\begin{aligned}
 z = \text{Maximizar} \quad & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\
 \text{Sujeito a:} \quad & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\
 & x_1, x_2, x_3, x_4 \in \{0, 1\}
 \end{aligned}$$

Os valores de $f_k(\lambda)$ e $p_k(\lambda)$ são mostrados na Tabela 2.1. Os valores de $f_0(\lambda)$ e de $f_k(0)$ são obtidos de imediato, enquanto que as outras colunas são preenchidas de cima para baixo utilizando a recursão $f_k(\lambda) = \max\{f_{k-1}(\lambda), c_k + f_{k-1}(\lambda - a_k)\}$. Por exemplo, $f_2(7) = \max\{f_1(7), 7 + f_1(7 - 1)\} = \max\{10, 7 + 10\} = 17$, e como o valor de $f_2(7)$ é dado pelo segundo termo da maximização, fazemos $p_2(7) = 1$. O valor ótimo é $z^* = f_4(7) = 34$.

Para obter a solução ótima x^* , basta percorrer a tabela P , iniciando em $p_4(7)$. Como $p_4(7) = 1$, $x_4^* = 1$. $p_3(7 - 5) = p_3(2) = p_2(2) = 0$ e portanto $x_3^* = x_2^* = 0$. $p_1(2) = 1$, e portanto $x_1^* = 1$. Sendo assim, a solução ótima é $x^* = (1, 0, 0, 1)$.

2.3 Linearização por Partes

Seja uma função contínua $f(y)$ e $p_i = (a_i, f(a_i))$ para $i = 0, \dots, k$ um conjunto de pontos conhecidos desta função, com $a_i < a_{i+1}$, $i = 0, \dots, k - 1$. Podemos aproximar $f(y)$ por um conjunto de

segmentos lineares, passando pelos pontos p_i , conforme ilustra a Figura 2.1. Qualquer função contínua de uma variável pode ser aproximada por uma função linear por partes, sendo a qualidade da aproximação dependente do tamanho dos segmentos lineares.

Seja $\tilde{f}(y)$ a aproximação linear de $f(y)$. A seguir, apresentamos três modelos que podem ser utilizados para representar $\tilde{f}(y)$ [13].

2.3.1 Modelo com pesos por ponto (Modelo I)

Nesta primeira abordagem, ilustrada pela Figura 2.1 são utilizadas variáveis reais (ou pesos) λ_i . Qualquer $a_0 \leq y \leq a_k$ pode ser escrito como:

$$y = \sum_{i=0}^k \lambda_i a_i, \quad \sum_{i=0}^k \lambda_i = 1, \quad \lambda = (\lambda_0, \dots, \lambda_k) \in \mathbb{R}_+^{k+1}.$$

Os λ_i não são únicos, mas se $a_i \leq y \leq a_{i+1}$ e $\lambda_i + \lambda_{i+1} = 1$ para algum $i \in \{0, \dots, k-1\}$, então obtemos $\tilde{f}(y) = \lambda_i f(a_i) + \lambda_{i+1} f(a_{i+1})$. Em outras palavras,

$$\tilde{f}(y) = \sum_{i=0}^k \lambda_i f(a_i), \quad \sum_{i=0}^k \lambda_i = 1, \quad \lambda \in \mathbb{R}_+^{k+1}.$$

Se pelo menos dois dos λ_i 's são positivos, e se λ_j e λ_k são positivos, então $k = j-1$ ou $k = j+1$. Esta condição pode ser modelada utilizando-se variáveis binárias x_i para $i = 1, \dots, k$ (com $x_i = 1$ se $a_{i-1} \leq y \leq a_i$ e $x_i = 0$ caso contrário), e as restrições

$$\begin{aligned} \lambda_0 &\leq x_1 \\ \lambda_i &\leq x_i + x_{i+1} \quad i = 1, \dots, k-1 \\ \lambda_k &\leq x_k \\ \sum_{i=1}^k x_i &= 1 \\ x &\in \mathbb{B}^k. \end{aligned}$$

Note que se $x_j = 1$, então $\lambda_i = 0$ para $\forall i \notin \{j-1, j\}$.

Funções lineares por partes que são convexas (côncavas) podem ser minimizadas (maximizadas) utilizando Programação Linear, pois as inclinações dos segmentos são crescentes (decrescentes). Porém, se a função não é côncava nem convexa, variáveis binárias são necessárias para selecionar o segmento correto para um dado valor de y [31].

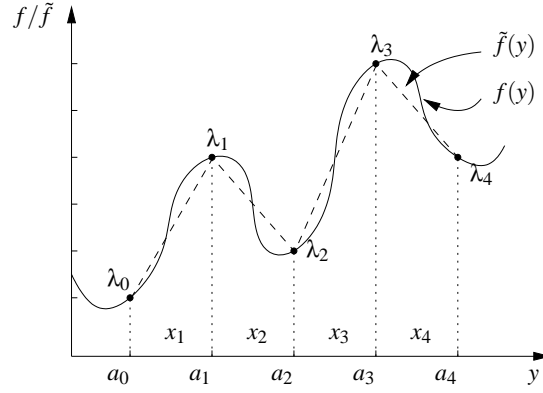


Figura 2.1: Linearização por partes para o modelo por pesos

2.3.2 Modelo com pesos sequencial (Modelo II)

Neste modelo, escrevemos y como:

$$y = a_0 + \lambda_1 + \dots + \lambda_k$$

onde $0 \leq \lambda_i \leq a_i - a_{i-1}$, $i = 1, \dots, k$.

Desejamos representar um dado y que pertence a uma região r da função. Então λ_i assume os valores:

$$\lambda_i = \begin{cases} a_i - a_{i-1} & \text{se } i < r \\ y - a_i & \text{se } i = r \\ 0 & \text{se } i > r \end{cases}$$

Variáveis inteiras x_i são utilizadas para selecionar as regiões anteriores a y . Assim, se considerarmos que y pertence a uma região r :

$$x_i = \begin{cases} 1 & \text{se } i < r \\ 0 & \text{se } i \geq r \end{cases}$$

Portanto, a formulação deste modelo pode ser escrita como:

$$y = a_0 + \sum_{i=1}^k \lambda_i \quad (2.4a)$$

$$\tilde{f}(y) = f(a_0) + \sum_{i=1}^k \frac{f(a_i) - f(a_{i-1})}{a_i - a_{i-1}} \lambda_i \quad (2.4b)$$

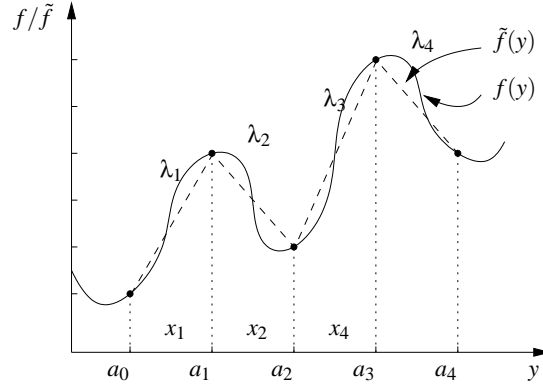


Figura 2.2: Linearização por partes para o modelo sequencial

com as seguintes restrições:

$$\lambda_1 \leq a_1 - a_0 \quad (2.5a)$$

$$\lambda_i \leq (a_i - a_{i-1})x_{i-1} \quad i = 2, \dots, k \quad (2.5b)$$

$$\lambda_i \geq (a_i - a_{i-1})x_i \quad i = 1, \dots, k-1 \quad (2.5c)$$

$$\lambda_k \geq 0 \quad (2.5d)$$

$$\lambda_i \in \mathbb{R} \quad i = 1, \dots, k \quad (2.5e)$$

$$x_i \in \mathbb{B} \quad i = 1, \dots, k-1 \quad (2.5f)$$

Note que em (2.5c) está implícita a condição $\lambda_i \geq 0$, pois $\lambda_i \geq (a_i - a_{i-1})x_i \geq 0$, onde $x_i \in \{0, 1\}$ e $a_i > a_{i-1}$. Observe também que em (2.5f) não é necessário definir x_k , pois se x pertencer à última região (k), $x_i = 1, \forall i \in \{1, \dots, k-1\}$.

Em [33] uma comparação entre os Modelos I e II é apresentada. Uma transformação do Modelo I para o espaço de variáveis de decisão do Modelo II é realizada, e prova-se que o Modelo II está estritamente contido nesta transformação. Além disso, este trabalho mostra que nos vértices do poliedro correspondente à relaxação linear do Modelo II as variáveis x são sempre inteiras. Ou seja, um problema de otimização cuja função objetivo seja linearizada utilizando o Modelo II pode ser resolvido por meio de programação linear. Neste aspecto, o Modelo II é considerado muito melhor que o Modelo I. Entretanto, sob o aspecto didático, o Modelo I é mais intuitivo que o II, devido ao significado de suas variáveis de decisão e simplicidade das restrições. Em [33], uma formulação que transforma o Modelo II no espaço de variáveis de decisão de I é proposta, mas suas restrições continuam pouco intuitivas. Então surge em [37] uma formulação simples com variáveis de decisão semelhantes às do Modelo I e com as características poliedrais do Modelo II, mostrada a seguir.

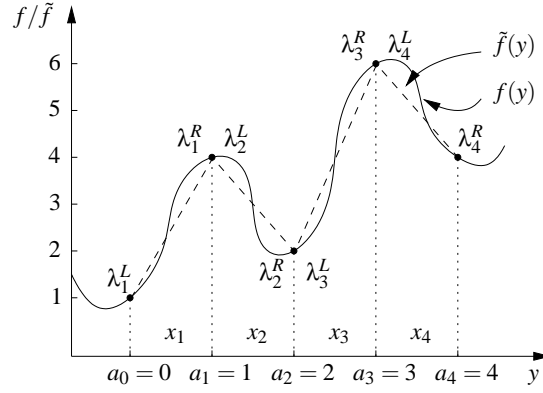


Figura 2.3: Linearização por partes para o modelo de Serali

2.3.3 Modelo de Serali (Modelo III)

Na formulação proposta em [37], são utilizadas duas variáveis de peso para cada região da linearização: λ_i^L é o peso do ponto à esquerda da região e λ_i^R é o peso do ponto à direita. Também são empregadas variáveis x_i inteiras que sequecionam a região. Esta formulação é mais geral que as anteriores, pois permite também linearizar funções descontínuas¹.

$$y = \sum_{i=1}^k [a_{i-1}\lambda_i^L + a_i\lambda_i^R] \quad (2.6a)$$

$$\tilde{f}(y) = \sum_{i=1}^k [f(a_{i-1})\lambda_i^L + f(a_i)\lambda_i^R] \quad (2.6b)$$

com restrições

$$\lambda_i^L + \lambda_i^R = x_i \quad i = 1, \dots, k \quad (2.7a)$$

$$\sum_{i=1}^k x_i = 1 \quad (2.7b)$$

$$\lambda_i^L, \lambda_i^R \geq 0 \quad i = 1, \dots, k \quad (2.7c)$$

$$\lambda_i^L, \lambda_i^R \in \mathbb{R} \quad i = 1, \dots, k \quad (2.7d)$$

$$x_i \in \mathbb{B} \quad i = 1, \dots, k \quad (2.7e)$$

Na Figura 2.3 vemos uma ilustração deste modelo. Para exemplificar, tomemos $y = 2,7$, que implica $i = 3$. De (2.7b) e (2.7e), $x_3 = 1$ e $x_1 = x_2 = x_4 = 0$. De (2.6a), (2.7a), (2.7c) e (2.7d), $\lambda_i^L = \lambda_i^R = 0$ para $i = 1, 2, 4$, $\lambda_3^L = 0,3$ e $\lambda_3^R = 0,7$.

¹Em [37] é mostrado que quando $f(y)$ é contínua, este modelo é equivalente ao modelo anterior.

Para calcular $\tilde{f}(y)$ utilizamos (2.6b): $\tilde{f}(y) = f(a_2)\lambda_3^L + f(a_3)\lambda_3^R = 2 \cdot 0,3 + 6 \cdot 0,7 = 4,8$.

Do exemplo apresentado, (2.7a) e (2.7b) são escritas matricialmente como $\hat{A}z = \hat{b}$,

$$\begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} z = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$z^T = \begin{bmatrix} \lambda_1^L & \lambda_1^R & x_1 & \lambda_2^L & \lambda_2^R & x_2 & \lambda_3^L & \lambda_3^R & x_3 & \lambda_4^L & \lambda_4^R & x_4 \end{bmatrix}$$

Desejamos mostrar que \hat{A} é totalmente unimodular. Podemos desconsiderar as colunas correspondentes às variáveis λ , pois estas possuem apenas uma linha diferente de zero [31]. A matriz resultante (colunas de x) corresponde a uma matriz de incidência [42], que é totalmente unimodular. Assim, a matriz \hat{A} também é totalmente unimodular.

As restrições (2.7c) a (2.7e), na forma $x_i \geq 0$ e $x_i \leq 1$ para todo $i = 1, \dots, k$, implicam em identidades que são concatenadas à matriz \hat{A} , e não afetam sua unimodularidade. Note também que as restrições de igualdade podem ser transformadas em desigualdades ($\pi^T z = \pi_0 \implies \pi^T z \geq \pi_0$ e $\pi^T z \leq \pi_0$). Assim, a matriz de restrições para esta formulação é totalmente unimodular.

A consequência é que o poliedro desta formulação é mínimo, ou seja, ele corresponde ao fecho convexo das soluções, e em todos os seus vértices as variáveis x são inteiras. Assim, problemas de otimização formulados de acordo com o Modelo III podem ser resolvidos utilizando programação linear.

2.4 Definições de Poliedro e Dimensão

Apresentamos aqui alguns resultados sobre Teoria Poliedral [20], [31], com o objetivo de familiarizar o leitor com este assunto.

Definição 2.1 Um conjunto de pontos $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ é linearmente independente se a única solução para $\sum_{i=1}^k \lambda_i x_i = 0$ é $\lambda_i = 0$ para $i = 1, 2, \dots, k$.

Definição 2.2 Um conjunto de pontos $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ é afim independente se a única solução para $\sum_{i=1}^k \lambda_i x_i = 0$ e $\sum_{i=1}^k \lambda_i = 0$ é $\lambda_i = 0$ para $i = 1, 2, \dots, k$.

Definição 2.3 Seja $S = \{x_1, x_2, \dots, x_k\}$ um conjunto de pontos em \mathbb{R}^n . O fecho convexo de S é o conjunto de pontos $\text{conv}(S) = \{\sum_{i=1}^k \lambda_i x_i : \sum_{i=1}^k \lambda_i = 1, x_i \in S, \lambda_i \in \mathbb{R} \text{ e } \lambda_i \geq 0, \text{ para } i = 1, 2, \dots, k\}$.

Definição 2.4 Um poliedro $P \subseteq \mathbb{R}^n$ é um conjunto de pontos que satisfaz um número finito de desigualdades lineares, ou seja, $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, onde (A, b) é uma matriz $m \times (n + 1)$.

Proposição 2.1 O fecho convexo dos elementos inteiros de $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ é também um poliedro, ou seja, $\text{conv}(P \cap \mathbb{Z}^n) = \{x \in \mathbb{R}^n : \tilde{A}x \leq \tilde{b}\}$ para alguma matriz \tilde{A} e algum vetor \tilde{b} .

Definição 2.5 Um poliedro $P \subseteq \mathbb{R}^n$ é limitado se existe $\omega \in \mathbb{R}_+^1$ tal que $P \subseteq \{x \in \mathbb{R}^n : -\omega \leq x_j \leq \omega \text{ para } j = 1, \dots, n\}$. Um poliedro limitado é denominado politopo.

Definição 2.6 Um poliedro P tem dimensão n , denotado por $\dim(P) = n$, se o número máximo de pontos afim independentes em P é $n + 1$.

Definição 2.7 Um poliedro $P \subseteq \mathbb{R}^n$ possui dimensão cheia se $\dim(P) = n$.

Se P não possui dimensão cheia, então pelo menos uma de suas desigualdades $a_i x \leq b_i$ é satisfeita na igualdade por todos os pontos de P , onde a_i corresponde à linha i da matriz A . Sejam $M = \{1, 2, \dots, m\}$, $M^= = \{i \in M : a_i x = b_i \text{ para todo } x \in P\}$, e $M^< = \{i \in M : a_i x < b_i \text{ para algum } x \in P\} = M \setminus M^=$. Sejam $(A^=, b^=)$, $(A^<, b^<)$ as linhas correspondentes de (A, b) , formando os conjuntos de igualdades e desigualdades da representação (A, b) de P , ou seja, $P = \{x \in \mathbb{R}^n : A^=x = b^=, A^<x \leq b^<\}$. Note que se $i \in M^<$, então (a_i, b_i) não pode ser escrito como combinação linear das linhas de $(A^=, b^=)$.

2.5 Descrição de Poliedros através de Facetas

Dado um poliedro $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, analisamos nesta seção quais das desigualdades $a_i x \leq b_i$ são necessárias para descrever P e quais podem ser descartadas.

Definição 2.8 A desigualdade $\pi^T x \leq \pi_0$ ou (π, π_0) é denominada uma desigualdade válida para P se for satisfeita por todos os pontos de P .

Definição 2.9 Se (π, π_0) é uma desigualdade válida para P e $F_\pi = \{x \in P : \pi^T x = \pi_0\}$, F_π é denominada uma face de P , e dizemos que (π, π_0) representa ou induz F_π . Uma face F_π é dita própria se $F_\pi \neq \emptyset$ e $F_\pi \neq P$.

A face F_π representada por (π, π_0) é não-vazia se e somente se $\max\{\pi^T x : x \in P\} = \pi_0$. Quando F_π é não-vazia, dizemos que (π, π_0) suporta P .

Definição 2.10 Uma face F de P é uma faceta de P se $\dim(F) = \dim(P) - 1$.

Teorema 2.1 Seja P um poliedro descrito por $\{x \in \mathbb{R}^n : Ax \leq b \text{ e } Cx = d\}$, onde $Ax \leq b$ não possui nenhuma igualdade implícita e C possui posto completo. Esta descrição do poliedro é não-redundante, ou seja, a remoção de uma desigualdade ou igualdade resulta em um poliedro diferente de P , se e somente se toda desigualdade $Ax \leq b$ define uma faceta para P .

Definição 2.11 Dizemos que uma desigualdade $\gamma^T x \leq \gamma_0$ domina outra desigualdade $\pi^T x \leq \pi_0$, ou seja, $\gamma^T x \leq \gamma_0$ é mais forte que $\pi^T x \leq \pi_0$, se existir $\alpha > 0$ tal que $\gamma \geq \alpha\pi$ e $\gamma_0 \leq \alpha\pi_0$.

Algumas vezes é possível obter desigualdades mais fortes, ou até mesmo facetas, a partir de faces de um poliedro P . Um processo bastante utilizado para este fim é o *lifting*. Considere a desigualdade válida $\pi^T x \leq \pi_0$ para $P \subseteq \mathbb{R}^n$, e F_π a face definida por esta desigualdade. Considere outra desigualdade válida para P , $\gamma^T x \leq \gamma_0$, que define a face F_γ . A desigualdade $\gamma^T x \leq \gamma_0$ é um *lifting* da desigualdade $\pi^T x \leq \pi_0$ se:

1. $F_\pi \subset F_\gamma$;
2. $\dim(F_\pi) < \dim(F_\gamma) \leq \dim(P) - 1$.

2.5.1 Caracterização de Facetas

Descrevemos agora duas técnicas utilizadas para provar que desigualdades válidas definem facetas para um dado poliedro: construção direta e verificação da maximalidade. Suponha que o poliedro $P = \{x \in \mathbb{R}^n : Ax \leq b\} \subseteq \mathbb{R}^n$ possui dimensão cheia, e que a desigualdade $\pi^T x \leq \pi_0$ pertence ao conjunto de desigualdades que definem P (portanto uma desigualdade válida).

Construção direta

Para provar que a desigualdade válida $\pi^T x \leq \pi_0$ define uma faceta para P utilizando a técnica da construção direta, é necessário encontrar um conjunto de vetores $S \subseteq P$ afim independentes com cardinalidade igual a $\dim(P)$, tal que cada elemento de S satisfaz a desigualdade $\pi^T x \leq \pi_0$ na igualdade. Assim, a face F definida por $\pi^T x = \pi_0$ teria dimensão igual a $\dim(P) - 1$ e portanto (π, π_0) definiria uma faceta para P . Caso o poliedro não possua dimensão cheia, primeiro deve-se verificar que $\pi^T x \leq \pi_0$ não é uma igualdade implícita. Isto pode ser feito encontrando-se um vetor $\hat{x} \in P$ tal que $\pi^T \hat{x} < \pi_0$.

Verificação da maximalidade

Esta técnica consiste em provar que a face F_π de $P \subseteq \mathbb{R}^n$ induzida por (π, π_0) não está contida em nenhuma outra face de P , ou seja, provar que F_π é máxima. Seja $\gamma^T x \leq \gamma_0$ uma desigualdade válida para P , tal que $F_\pi \subseteq F_\gamma$. Se $\gamma^T x \leq \gamma_0$ pode ser expressa como um múltiplo escalar não negativo de $\pi^T x \leq \pi_0$, então (π, π_0) define uma faceta para P . Caso P não possua dimensão cheia, deve-se provar que (π, π_0) não é uma igualdade implícita, e também garantir que (γ, γ_0) e (π, π_0) diferem não apenas de um múltiplo escalar positivo, mas também de uma combinação linear das equações implícitas para P .

2.6 Desigualdades Válidas para o Problema da Mochila

Utilizaremos aqui a mesma instância do problema da mochila apresentada na Seção 2.2. A relaxação linear de KP é apresentada abaixo:

$$\begin{aligned} LKP: \quad z = \text{Maximizar} \quad & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\ \text{Sujeito a:} \quad & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\ & 0 \leq x \leq 1 \end{aligned}$$

Resolvendo LKP , obtemos a solução ótima $x = (1, 1, 0, 0.8)$, com valor objetivo $z = 36.2$. A desigualdade $x_1 + x_2 + x_4 \leq 2$ é válida para KP , pois somente 2 itens do conjunto $\{1, 2, 4\}$ cabem na mochila ao mesmo tempo (a mochila possui capacidade 7, e os pesos dos itens 1, 2 e 4 são 2, 1 e 5 respectivamente). Ao introduzir esta desigualdade na formulação, a solução ótima encontrada para LKP é cortada, pois $x_1 + x_2 + x_3 = 2.8 > 2$. Sendo assim, obtemos um novo problema:

$$\begin{aligned} LKP_1: \quad z = \text{Maximizar} \quad & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\ \text{Sujeito a:} \quad & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\ & x_1 + x_2 + x_4 \leq 2 \\ & 0 \leq x \leq 1 \end{aligned}$$

A solução de LKP_1 é $x = (0, 1, 0.1667, 1)$, com valor objetivo $z = 35.1667$. Seguindo o mesmo raciocínio anterior, obtemos a desigualdade $x_2 + x_3 + x_4 \leq 2$, que corta a solução fracionária obtida, pois $x_2 + x_3 + x_4 = 2.1667 > 2$. Introduzimos esta desigualdade válida em KP e obtemos a nova relaxação linear:

$$\begin{aligned} LKP_2: \quad z = \text{Maximizar} \quad & 10x_1 + 7x_2 + 25x_3 + 24x_4 \\ \text{Sujeito a:} \quad & 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\ & x_1 + x_2 + x_4 \leq 2 \\ & x_2 + x_3 + x_4 \leq 2 \\ & 0 \leq x \leq 1 \end{aligned}$$

cuja solução é $x = (0.1429, 0.8571, 0.1429, 1)$, com valor objetivo $z = 35$. A desigualdade $x_1 + x_2 + x_3 + x_4 \leq 2$ é válida² para KP e corta a solução obtida, pois $x_1 + x_2 + x_3 + x_4 = 2.1429 > 2$. Introdu-

²Note que esta desigualdade domina as desigualdades inseridas em LKP_1 e LKP_2 .

zimos esta desigualdade em KP e obtemos uma nova relaxação linear:

$$\begin{array}{ll}
 LKP_3 : & z = \text{Maximizar} \quad 10x_1 + 7x_2 + 25x_3 + 24x_4 \\
 & \text{Sujeito a:} \quad 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\
 & \quad \quad \quad x_1 + x_2 \quad \quad \quad + x_4 \leq 2 \\
 & \quad \quad \quad \quad \quad x_2 + x_3 + x_4 \leq 2 \\
 & \quad \quad \quad x_1 + x_2 + x_3 + x_4 \leq 2 \\
 & \quad \quad \quad 0 \leq x \leq 1
 \end{array}$$

A solução obtida é $x = (0, 0.8, 0.2, 1)$, com valor objetivo $z = 34.6$. A desigualdade $x_3 + x_4 \leq 1$ é válida para KP e corta a solução obtida, pois $x_3 + x_4 = 1.2 > 1$. Introduzimos esta desigualdade em KP e obtemos a nova relaxação linear:

$$\begin{array}{ll}
 LKP_4 : & z = \text{Maximizar} \quad 10x_1 + 7x_2 + 25x_3 + 24x_4 \\
 & \text{Sujeito a:} \quad 2x_1 + x_2 + 6x_3 + 5x_4 \leq 7 \\
 & \quad \quad \quad x_1 + x_2 \quad \quad \quad + x_4 \leq 2 \\
 & \quad \quad \quad \quad \quad x_2 + x_3 + x_4 \leq 2 \\
 & \quad \quad \quad x_1 + x_2 + x_3 + x_4 \leq 2 \\
 & \quad \quad \quad \quad \quad \quad \quad x_3 + x_4 \leq 1 \\
 & \quad \quad \quad 0 \leq x \leq 1
 \end{array}$$

cuja solução é $x = (1, 0, 0, 1)$, com valor objetivo $z = 34$. Como a solução obtida é inteira, conclui-se que é a solução ótima para KP , ou seja, $x^* = (1, 0, 0, 1)$ e $z^* = 34$ (a mesma solução obtida no Capítulo 2.2).

A introdução de desigualdades válidas é bastante utilizada nos programas comerciais de otimização, além do pré-processamento da formulação (eliminar variáveis e restrições redundantes, de modo a obter uma formulação mais compacta, que tipicamente pode ser resolvida mais rapidamente). A idéia é examinar a formulação inicial $P = \{x : Ax \leq b, x \geq 0\}$ com $X = P \cap \mathbb{Z}^n$, encontrar um conjunto de desigualdades válidas $Qx \leq q$ para X , adicionar estas desigualdades na formulação original e obter a nova formulação $P' = \{x : Ax \leq b, Qx \leq q, x \geq 0\}$ com $X = P' \cap \mathbb{Z}^n$. Caso a relaxação linear de P' não forneça uma solução inteira, podemos tentar encontrar a solução ótima através de um algoritmo de otimização para P' , que é uma formulação mais “apertada”.

Um algoritmo exato bastante utilizado para este fim é chamado de *Branch-and-Bound* ($B\&B$). Um algoritmo $B\&B$ pode ser entendido como uma estratégia de divisão de um problema P em um conjunto de subproblemas $\{SP_k\}$ de forma que a solução de P possa ser obtida através da combinação das soluções dos subproblemas. De acordo com o algoritmo $B\&B$, as divisões são feitas iterativamente, sempre observando que os subproblemas devem ser mais fáceis de serem resolvidos do que o problema original, além de descartar subproblemas por meio de enumeração implícita, ou seja, garantir de alguma forma que a solução ótima não é solução de um certo subproblema, e que portanto pode-

mos descartá-lo. Na técnica conhecida como *Cut-and-Branch*, cortes são inseridos iterativamente na relaxação até que novos cortes não possam ser gerados, e então é realizada uma nova busca *B&B* na formulação resultante. Quando a geração de cortes é executada em algum nó da árvore de *B&B*, o método recebe o nome de *Branch-and-Cut*.

2.7 Sumário

Neste capítulo foram apresentados conceitos básicos da área de otimização e o paradigma de Programação Dinâmica (PD), cujo funcionamento foi ilustrado através de sua aplicação para a solução de uma instância de um problema clássico de otimização conhecido como “problema da mochila” [22]. Foram apresentados também conceitos de funções lineares por partes e teoria poliedral. Apresentamos um exemplo numérico de como as desigualdades válidas de um problema podem ser utilizadas para facilitar (e em alguns casos encontrar) a solução do mesmo.

Capítulo 3

Solução de $P(\emptyset)$ Utilizando Programação Dinâmica

Neste capítulo, apresentamos um modelo do problema de alocação de gás de elevação em linguagem formal, e um algoritmo de Programação Dinâmica que resolve este problema. Apresentamos também resultados de experimentos numéricos que comprovam a eficiência do algoritmo em termos de tempo computacional e de qualidade da solução encontrada. Algumas comparações com exemplos publicados na literatura mostram que o desempenho de nosso algoritmo supera o desempenho das soluções propostas até o momento.

3.1 Formulação do Problema

O problema estudado aqui visa encontrar a alocação ótima de uma taxa disponível de injeção de gás de elevação para um grupo de poços, visando maximizar o lucro obtido com a venda dos hidrocarbonetos extraídos. Este problema pode ser escrito da seguinte maneira:

$$\begin{aligned} P(\emptyset): \quad J = \quad & \text{Maximizar} \quad f = \sum_{n=1}^N (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) q_p^n - \sum_{n=1}^N p_i q_i^n \\ & \text{Sujeito a:} \\ & q_p^n = \tilde{q}_p^n(q_i^n) \quad n = 1, \dots, N \\ & \sum_{n=1}^N q_i^n \leq q_i^{\max} \\ & l_n y_n \leq q_i^n \leq u_n y_n \quad n = 1, \dots, N \\ & y_n \in \{0, 1\} \quad n = 1, \dots, N \end{aligned} \tag{3.1}$$

onde:

- N é o número de poços que produzem via *gas-lift*;
- q_i^n é a taxa de injeção de gás alocada para o poço n ;
- y_n assume valor 1 se o poço n está ativo, e 0 caso contrário;
- q_p^n é a taxa de produção de fluidos do poço n , em função de q_i^n ;
- q_i^{max} é a taxa de injeção de gás disponível, sendo que a soma das taxas de injeção de gás dos N poços não deve ultrapassar esta taxa;
- l_n e u_n são as taxas de injeção mínima e máxima para o poço n , respectivamente;
- p_o e p_g representam o lucro (preço de venda menos custos de processamento) obtido por volume de óleo e gás vendido, respectivamente;
- p_w é o custo de tratamento da água¹;
- p_i representa os custos de compressão, tratamento e injeção do gás;
- γ_o^n , γ_g^n e γ_w^n são respectivamente as frações de óleo, gás e água² que compõe o fluxo de saída do poço n , sendo $\gamma_o^n + \gamma_g^n + \gamma_w^n = 1$.

O problema $P(\emptyset)$ pode ser facilmente adaptado para capturar algumas variações:

- A não-concavidade introduzida por um poço n que não responde imediatamente à injeção de gás ($q_p^n(q_i^n) = 0$ para algum $q_i^n > 0$, poços dos tipos C e D , Figura 1.4) é automaticamente eliminada devido à existência do limite inferior de injeção l_n ;
- Neste trabalho assumimos que $q_p^n(0) = 0$ para $n = 1, \dots, N$. No caso de existir um poço surgente, ou seja, um poço para o qual $q_p^n(0) \neq 0$ (poços do tipo A , Figura 1.4), basta adicionar à função objetivo sua contribuição $J_s^n = (p_o\gamma_o^n + p_g\gamma_g^n - p_w\gamma_w^n)q_p^n(0)$, e substituir a função $q_p^n(q_i^n)$ por $\hat{q}_o^n(q_i^n) = q_p^n(q_i^n) - q_p^n(0)$; a solução ótima q_i^n para $n = 1, \dots, N$ não se altera com a adição da constante J_s^n no valor de J e com a utilização de $\hat{q}_o^n(q_i^n)$, e portanto os algoritmos desenvolvidos aqui são capazes de levar estes poços em consideração;
- Caso o problema de interesse seja a maximização da taxa de produção de óleo ao invés do lucro obtido, basta fazer $p_o = 1$, e $p_g = p_w = p_i = 0$.

¹O descarte da água só pode ser feito dentro de determinadas especificações, regulamentadas por órgãos de controle do meio ambiente, limitando a quantidade de poluentes como o teor de óleo e H_2S nos efluentes aquosos [39].

²Se considerarmos que a pressão e a temperatura no separador forem constantes, podemos considerar estas frações também constantes para um determinado poço.

Para completar a descrição do problema de otimização em Programação Matemática, é necessário definir a relação entre q_p^n e q_i^n . O problema de maximização não pode ser resolvido se os dados que caracterizam a produção dos poços via *gas-lift*, ou seja, as funções $q_p^n = q_p^n(q_i^n)$ não estiverem disponíveis [15], [18], [19], [26]. Estes dados recebem o nome de curvas de desempenho (*Well Performance Curves*, ou *WPCs*), e tipicamente possuem a forma mostrada na Figura 1.3. Os dados para a construção das *WPCs* são obtidos através de simulações e testes de produção: aplica-se um determinado número de taxas de injeção de gás em um determinado poço n , e o seu fluxo de saída é direcionado a um separador de testes, obtendo assim um conjunto de k pontos $Q^n = \{(q_i^{n,1}, q_p^{n,1}), (q_i^{n,2}, q_p^{n,2}), \dots, (q_i^{n,k}, q_p^{n,k})\}$, bem como as porcentagens de óleo, gás e água produzidas pelo poço, γ_o^n , γ_g^n e γ_w^n . Dado um conjunto Q^n de pontos de desempenho de *gas-lift*, com $q_i^{n,j} < q_i^{n,j+1}$ para $j = 1, \dots, k-1$, foram utilizados três modelos de curvas³ para representar as *WPCs*:

- 1) Uma função polinomial da forma $\tilde{q}_p^n(q_i^n) = \alpha_0^n y_n + \alpha_1^n q_i^n + \alpha_2^n (q_i^n)^2 + \alpha_3^n (q_i^n)^3$, definida pelas constantes α_0^n , α_1^n , α_2^n e α_3^n . A vantagem deste modelo é que o problema de ajuste de curvas gerado é um problema de fácil solução. Sua desvantagem é a má qualidade de extrapolação para pontos fora do intervalo $[q_i^{n,1}, q_i^{n,k}]$;
- 2) Uma função exponencial da forma $\tilde{q}_p^n(q_i^n) = A_n(2 - e^{-B_n q_i^n}) - C_n e^{D_n q_i^n} + F_n(1 - y_n)$, definida pelas constantes A_n , B_n , C_n , D_n e $F_n = C_n - A_n$. Esta função exponencial é uma boa aproximação para o comportamento típico de uma *WPC*, fornece bons dados de extrapolação, e sua concavidade é garantida satisfazendo condições simples. Basta garantir que as constantes A_n e C_n sejam positivas, pois sob estas condições sua derivada segunda assume valores negativos para qualquer valor de q_i^n ;
- 3) Uma função logarítmica da forma $q_p^n(q_i^n) = c_1^n y_n + c_2^n q_i^n + c_3^n (q_i^n)^2 + c_4^n \ln(q_i^n + 1)$, definida pelas constantes c_1^n , c_2^n , c_3^n e c_4^n . Este modelo foi utilizado para comparar os resultados obtidos com o algoritmo PD e os obtidos com o método de SQP proposto em [1].

Hipótese 3.1 Cada função $q_p^n(q_i^n)$ é uma função côncava tal que $q_p^n(q_i^n) \geq 0$ para $q_i^n \in [l_n, u_n]$.

Hipótese 3.2 $p_o \gamma_o^n + p_g \gamma_g^n > p_w \gamma_w^n$ para todo poço n , o que implica em $\gamma_n = p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n > 0$.

A Hipótese 3.1 é válida pois normalmente o comportamento do fluxo de saída de um poço em função de sua taxa de injeção de gás possui o aspecto mostrado na Figura 1.3, ou seja, a função $q_p(q_i)$ é crescente até uma determinada taxa de saturação, quando os efeitos friccionais passam a ditar o comportamento do fluxo, e $q_p(q_i)$ passa a ser decrescente. A Hipótese 3.2 reflete o aspecto da viabilidade econômica da exploração de um poço de petróleo, ou seja, caso o custo de tratamento da água extraída supere os dividendos provenientes da venda dos hidrocarbonetos, o poço é economicamente inviável, e portanto a hipótese é válida.

³Os métodos de ajuste de curvas utilizados para ajustar os modelos aos pontos em Q_n encontram-se no Apêndice A.

Os valores de γ_o^n , γ_g^n e γ_w^n podem ser escritos em termos da razão água-óleo (ou *Water Cut*, WC) e da razão gás-líquido (ou *Gas-Liquid Ratio*, GLR), termos mais comuns na literatura. Definindo WC e GLR como:

$$GLR = \frac{\gamma_g}{\gamma_o + \gamma_w}, \quad WC = \frac{\gamma_w}{\gamma_o}$$

e sabendo que $\gamma_o^n + \gamma_g^n + \gamma_w^n = 1$, obtemos as seguintes relações:

$$\gamma_o = \frac{1}{(1+WC)(1+GLR)}, \gamma_w = \frac{WC}{(1+WC)(1+GLR)}, \gamma_g = \frac{GLR}{1+GLR}.$$

Corolário 3.1 f é uma função côncava de $q_i = [q_i^1, \dots, q_i^N]$ na região factível de $P(\emptyset)$.

Prova Como q_p^n é uma função côncava (Hipótese 3.1) e γ_n é uma constante positiva para cada n (Hipótese 3.2), $f_n = \gamma_n q_p^n - p_i q_i^n$ é uma função côncava e portanto $f = f_1 + \dots + f_n$ é uma função côncava de q_i . ■

Proposição 3.1 $P(\emptyset)$ é NP-difícil.

Prova: O problema da mochila (*knapsack problem*, KP) é um problema NP-difícil e pode ser reduzido a $P(\emptyset)$. Seja I_{KP} uma instância de KP consistindo de N itens, onde c_n é o valor do n -ésimo item e w_n é o seu peso, e b é a capacidade da mochila. Gerando uma instância I de $P(\emptyset)$ com $q_i^{max} = b$, $l_n = u_n = w_n$ para todo n , $\gamma_o^n = 1$, $\gamma_g^n = \gamma_w^n = 0$, $p_o = 1$ e $p_g = p_w = p_i = 0$, $q_p^n = y_n c_n$ para cada n (para a função polinomial, basta fazer $\alpha_0^n = c_n$ e $\alpha_1^n = \alpha_2^n = \alpha_3^n = 0$; para a função exponencial, $A_n = c_n$ e $B_n = C_n = D_n = 0$; para a função logarítmica, $c_1 = c_n$ e $c_2 = c_3 = c_4 = 0$), uma solução ótima $z = \{y_n, q_i^n, q_p^n\}$ para I é também uma solução ótima para I_{KP} , ou seja, o n -ésimo item vai para a mochila se e somente se $y_n = 1$. ■

3.2 Solução de $P(\emptyset)$

Apresentamos aqui um algoritmo para resolver $P(\emptyset)$. Nesta seção, refere-se a $P(\emptyset)$ como P , por simplicidade. Como mostrado na Proposição 3.1, o problema P pertence à classe de problemas NP-difíceis, o que significa que até hoje não existe e as evidências são de que não existirá algoritmo capaz de fornecer uma solução para P em tempo polinomial no número de poços. Portanto, é necessário desenvolver um algoritmo que seja capaz de fornecer uma solução aproximada, próxima da solução ótima, e que o tempo exigido para o cálculo desta solução seja relativamente pequeno.

Para resolver P através de PD seria necessário resolver um número infinito de problemas, pois q_i é uma variável contínua [11]. Aqui, apresentamos um algoritmo PD que resolve P_M , uma versão aproximada de P , onde para cada poço n são alocadas w_n unidades padrão de injeção de gás. Seja $\delta = q_i^{max}/M$ uma unidade padrão de taxa de injeção de gás, onde M é o número total de unidades

padrão disponíveis. O problema P_M maximiza o lucro obtido, e é definido como:

$$\begin{aligned}
 P_M : J_M = & \text{ Maximizar } \sum_{n=1}^N [(p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n] \\
 \text{Sujeito a :} & \\
 & l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \quad n = 1, \dots, N \\
 & \sum_{n=1}^N w_n \leq M \\
 & y_n \in \{0, 1\} \quad n = 1, \dots, N \\
 & w_n \in \{0, \dots, M\} \quad n = 1, \dots, N
 \end{aligned} \tag{3.2}$$

onde a função \tilde{q}_p^n é a função que aproxima o comportamento da produção de fluidos do poço n em função de q_i^n e w_n é o número de unidades padrão de taxa de injeção de gás alocado para o poço n .

A seguir desenvolvemos as recorrências para a solução de P_M via PD, apresentamos o algoritmo para a solução do problema e os resultados numéricos obtidos.

3.2.1 Recorrências

Limitando a taxa de injeção de gás disponível em $m \in \{0, \dots, M\}$ unidades padrão, o problema assume a seguinte formulação recursiva:

$$\begin{aligned}
 \tilde{P}_{m,n} : F_{m,n} = & \text{ Maximizar } (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n \\
 \text{Sujeito a :} & \\
 & l_n y_n \leq q_i^n \leq \min\{u_n, m \delta\} y_n \\
 & y_n \in \{0, 1\}
 \end{aligned} \tag{3.3}$$

$$P_{m,N} : J_{m,N} = F_{m,N}$$

$$\begin{aligned}
 P_{m,n} : J_{m,n} = & \text{ Maximizar } (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n + J_{m-w_n, n+1} \\
 \text{Sujeito a :} & \\
 & l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \\
 & y_n \in \{0, 1\} \\
 & w_n \in \{0, \dots, m\}
 \end{aligned} \tag{3.4}$$

3.2.2 Algoritmo PD para $P(\emptyset)$

Das recorrências (3.3) e (3.4), conclui-se que $P_M \equiv P_{M,1}$ e consequentemente $J_M \equiv J_{M,1}$. A definição destas recorrências torna possível a construção de um algoritmo PD:

Algoritmo PD para P_M

Tabela 3.1: Parâmetros de uma instância de $P(\emptyset)$ com 6 poços.

	n					
	1	2	3	4	5	6
α_0^n	0	0	0	0	0	0
α_1^n	42,2210	47,1210	45,7210	38,9210	37,7210	39,4210
α_2^n	0	0	0	0	0	0
α_3^n	-0,2549	-0,2649	-0,2749	-0,2449	-0,2549	-0,2649
γ_o^n	0,70	0,75	0,65	0,65	0,60	0,80
γ_g^n	0,20	0,17	0,25	0,20	0,30	0,12
γ_w^n	0,10	0,08	0,10	0,15	0,10	0,08
l_n	3,65	3,65	3,65	3,65	3,65	3,65
u_n	10,00	10,00	10,00	10,00	10,00	10,00

Inicialização

Para $m = 0, \dots, M$

Calcule $J_{m,N}$

Recorrência

Para $n = N - 1, \dots, 1$

Para $m = 0, \dots, M$

Calcule $J_{m,n} = \max \{F_{w_n,n} + J_{m-w_n,n+1} : w_n = 0, 1, \dots, m\}$

O algoritmo resolve P_M em tempo $\Theta(NM^2)$ e demanda $\Theta(NM)$ unidades de memória para armazenar a tabela de valores $J_{m,n}$. Esta tabela contém as soluções ótimas para uma família de problemas, para $q_i^{disp} = 0, \delta, 2\delta, \dots, q_i^{max}$, onde q_i^{disp} é a taxa de injeção de gás disponível. Esta característica do algoritmo possui grande apelo prático, pois uma situação comum em campos de produção de petróleo são os casos de falha de um ou mais compressores de gás [23], o que faz com que a taxa de injeção disponível diminua. Neste caso, uma solução quase-ótima para esta nova situação está prontamente disponível. O algoritmo também pode armazenar uma tabela auxiliar $I_{m,n}$ para registrar os valores ótimos de q_i^n , semelhante ao indicador $p_k(\lambda)$ do exemplo da Seção 2.2.1.

Para ilustrar o funcionamento do algoritmo, mostraremos a solução de um exemplo numérico simples. A Tabela 3.1 mostra os parâmetros utilizados. A instância consiste de 6 poços e utilizamos o modelo polinomial $\tilde{q}_p^n(q_i^n) = \alpha_0^n + \alpha_1^n q_i^n + \alpha_2^n (q_i^n)^2 + \alpha_3^n (q_i^n)^3$ para ajustar as WPCs (modelo 1, Apêndice A). Utilizamos também $q_i^{max} = 40$, $p_o = 1$, $p_g = 0,6$, $p_w = 0,1$ e $p_i = 0,05$. A Tabela 3.2 contém os valores da função objetivo, e a Tabela 3.3 contém os valores ótimos de taxa de injeção de gás. Note que a solução ótima é dada por $J_{M,1}$.

Tabela 3.2: Tabela $J_{m,n}$, com destaque para a solução ótima $J_{M,1} = J_{10,1}$.

m	$m\delta$	n					
		1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	4	144,5716	144,5716	130,3794	121,3910	121,3910	121,3910
2	8	274,9510	274,9510	251,7704	226,8989	224,8102	159,5714
3	12	398,3330	396,3421	357,2783	330,3181	262,9907	159,5714
4	16	519,7241	501,8499	460,6975	368,4985	295,2172	159,5714
5	20	625,2320	605,2691	509,2381	405,2128	295,2172	159,5714
6	24	728,6512	664,4729	547,4186	437,4394	295,2172	159,5714
7	28	787,8549	713,0135	584,1329	437,4394	295,2172	159,5714
8	32	836,3956	751,1940	616,3594	437,4394	295,2172	159,5714
9	36	882,0529	787,9083	616,3594	437,4394	295,2172	159,5714
10	40	920,2333	820,1349	616,3594	437,4394	295,2172	159,5714

3.3 Resultados Numéricos

Deseja-se agora verificar quão bem P_M aproxima P e mostrar que o algoritmo é suficientemente rápido para a aplicação em questão. Foram realizados 60 experimentos, variando o número de poços ($N = 6, 12, 24$ e 48), o nível de discretização ($M = 10, 50, 100, 200, 500$), e a taxa de injeção disponível (três valores distintos de q_i^{max}). Os valores objetivo J_M encontrados pelo algoritmo em cada teste foram comparados a um limite superior UB encontrado através de relaxação contínua e programação côncava ⁴ (*MINOS solver version 5.51* [27]).

A Tabela 3.4 mostra os resultados dos testes de qualidade de aproximação, onde a primeira coluna contém os níveis de discretização, a segunda coluna representa a porcentagem média⁵ do limite superior UB alcançada pela solução J_M ($J_M \times 100/UB$), e a terceira coluna mostra o tempo de execução médio do algoritmo. Observa-se que a solução encontrada se aproxima bastante do limite superior UB , a não ser quando o nível de discretização é muito pequeno ($M = 10$), e que o tempo de execução do algoritmo é relativamente baixo, alcançando a marca de 2 minutos para $M = 500$. Porém, utilizando um nível de discretização $M = 100$, o tempo de execução cai para 5 segundos, enquanto a qualidade da solução permanece praticamente a mesma. Em média, a solução levou 29,87s para ser encontrada, e ficou a 5,94% do limite superior. Se desconsiderarmos o nível de discretização mais grosseiro ($M = 10$), o tempo de solução médio sobe para 37,32s e a distância média da solução para o limite superior cai para apenas 0,72%. O algoritmo foi implementado nas linguagens de programação

⁴A função que relaciona o fluxo de saída do poço e sua taxa de injeção de gás é uma função côncava no intervalo $[0, u_n]$, $n = 1, \dots, N$, garantindo assim a concavidade do problema quando a integralidade das variáveis y é relaxada. Isto permite encontrar a solução ótima global da relaxação contínua, ou seja, um limite superior.

⁵Média das soluções das instâncias de mesmo M e diferentes N e q_i^{max} .

Tabela 3.3: Tabela auxiliar $I_{m,n}$, com destaque para a alocação ótima de gás $q_i = (7,4251; 7,6954; 7,4406; 4,0000; 4,0000; 7,0379)$.

$m\delta$	q_i^1	q_i^2	q_i^3	q_i^4	q_i^5	q_i^6
0	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
4	0,0000	4,0000	4,0000	0,0000	0,0000	4,0000
8	0,0000	4,0000	4,0000	4,0000	4,0000	7,0379
12	4,0000	4,0000	4,0000	4,0000	4,0000	7,0379
16	4,0000	4,0000	4,0000	4,0000	7,0173	7,0379
20	4,0000	4,0000	7,4406	7,2722	7,0173	7,0379
24	4,0000	7,6954	7,4406	7,2722	7,0173	7,0379
28	4,0000	7,6954	7,4406	7,2722	7,0173	7,0379
32	4,0000	7,6954	7,4406	7,2722	7,0173	7,0379
36	7,4251	7,6954	7,4406	7,2722	7,0173	7,0379
40	7,4251	7,6954	7,4406	7,2722	7,0173	7,0379

Tabela 3.4: Qualidade média da solução e tempo médio de execução do algoritmo com relação ao nível de discretização M .

M	J_M/UB (%)	Tempo de execução (s)
10	73,14	0,10
50	98,80	1,32
100	99,35	4,66
200	99,47	21,57
500	99,52	121,72
Média	94,06	29,87

C/C++, em uma estação de trabalho equipada com processador Intel Pentium III 800MHz, 196MB de memória RAM, e sistema operacional GNU Linux.

Portanto, conclui-se que o algoritmo PD é capaz de fornecer soluções para o problema de alocação bem próximas da solução ótima teórica, e que o seu tempo de execução é baixo. Uma característica importante do algoritmo é que o seu tempo de execução cresce linearmente com o número de poços N . Uma possível deficiência seria o comportamento quadrático em M , que pode ser resolvido distribuindo a execução do algoritmo entre vários computadores. A Programação Dinâmica é uma técnica naturalmente paralelizável, sendo que se a execução do algoritmo for distribuída entre T unidades de processamento, seu tempo de execução diminuirá T vezes. Mesmo quando não se dispõe de várias unidades de processamento, os experimentos mostraram que M não necessita ser aumentado indefinidamente. Os resultados mostram que existe uma relação de compromisso entre a qualidade da solução (que está intimamente relacionada ao nível de discretização M) e o tempo de execução do algoritmo. Para as instâncias consideradas nos experimentos numéricos realizados, foi

Tabela 3.5: Resultados da alocação de gás de elevação, *Economic Slope* [18] e PD

	q_p (B/D)	q_i^{max} (MMSCF/D)	Ganho com PD (%)
<i>Individual Economic Slope</i>	10.004	3,33	
PD	11.081	3,33	+10,77
<i>Average Economic Slope</i>	10.350	3,58	
PD	11.134	3,58	+7,57
<i>Economic Slope</i> (gás limitado)	9.335	2,50	
PD	10.782	2,50	+15,50

possível encontrar soluções muito próximas (menos de 1%) da solução ótima teórica com um nível de discretização $M=100$ em um tempo bastante baixo, considerando os tempos altos (da ordem de horas ou dias) normalmente envolvidos no ramo de extração de petróleo.

3.3.1 Comparações com Resultados Publicados

O desempenho do algoritmo PD é agora comparado a quatro exemplos publicados [18], [6], [40], [1] e um *software* comercial [36]. Utilizamos os dados de WPCs fornecidos nestes trabalhos, e aplicamos o algoritmo PD desenvolvido aqui para encontrar a alocação ótima das taxas de injeção de gás. Nesta seção, MSCF/D é uma abreviação para *mil pés cúbicos padrão por dia*, e MMSCF/D é uma abreviação para *milhões de pés cúbicos padrão por dia*, medidas comuns para volume de gás injetado (as condições padrão são normalmente 60°F e 14.7 psia), e B/D significa *barris por dia*.

Em [18], o problema da alocação de gás de elevação foi resolvido utilizando as técnicas de “*individual economic slope*” e “*average economic slope*”, para situações de taxa de injeção de gás disponível ilimitada e limitada, considerando 6 poços. A Tabela 3.5 mostra a comparação das soluções obtidas através de PD com as soluções de [18]. Os resultados mostram que o algoritmo PD possui melhor desempenho em todos os três cenários.

Em [6], o problema de alocação foi resolvido utilizando os métodos “*equal-slope*” e “*Ex-In*”, para conjuntos de 5, 6 e 56 poços. A Tabela 3.6 mostra os resultados obtidos em [6] e utilizando PD. Analisando estes resultados, observa-se que o algoritmo PD possui um melhor desempenho em todos os testes.

Em [40] foram utilizados os dados fornecidos em [6], e um algoritmo de Programação Separável (*Separable Programming*, ou SP) foi aplicado para otimizar a produção de óleo de um conjunto de 56 poços, com 22.500 MMSCF/D de taxa de injeção de gás disponível. Os resultados obtidos em [40] são comparados aos obtidos através de PD (Tabela 3.7), e novamente o algoritmo desenvolvido no presente trabalho mostra um melhor desempenho.

Tabela 3.6: Resultados da alocação de gás de elevação, *Equal Slope*, *Ex-In* [6] e PD

	Poços	q_p (B/D)	q_i^{max} (MSCF/D)	Ganho com PD (%)
<i>Ex-In</i>	5	3.369,5	2.995,3	
PD	5	3.396,0	2.995,3	+0,79
<i>Equal Slope</i>	5	3.400,3	3.000,0	
PD	5	3.405,0	3.000,0	+0,16
<i>Ex-In</i>	6	3.629,0	4.427,7	
PD	6	3.634,6	4.427,7	+0,15
<i>Equal Slope</i>	6	3.510,2	4.373,7	
PD	6	3.628,3	4.373,7	+3,36
<i>Ex-In</i>	56	21.789,9	20.453,9	
PD	56	23.325,7	20.453,9	+7,05
<i>Equal Slope</i>	56	21.265,0	22.508,0	
PD	56	23.600,7	22.508,0	+10,99

Tabela 3.7: Resultados da alocação de gás de elevação, *Separable Programming* [40] e PD

	q_p (B/D)	q_i^{max} (MSCF/D)	Ganho com PD (%)
SP	22.632	22.500	
PD	23.378	22.500	+3,30

Para comparar os algoritmo PD e o método de SQP proposto em [1], o ajuste das curvas *WPCs* foram realizados com funções da forma $q_p^n(q_i^n) = c_1^n y_n + c_2^n q_i^n + c_3^n (q_i^n)^2 + c_4^n \ln(q_i^n + 1)$. Adaptar o algoritmo PD para trabalhar com estas funções é relativamente fácil, o que mostra que o algoritmo pode ser modificado para tratar diferentes métodos de ajuste de curvas. Utilizando o método de mínimos quadrados, as curvas logarítmicas foram ajustadas aos dados de desempenho contidos em [6] e [18], obtendo assim as instâncias do problema utilizadas nesta comparação. Os resultados obtidos com SQP e PD para os cenários onde não há poços com taxa de partida são mostrados na Tabela 3.8, e a Tabela 3.9 contém os resultados para os cenários com poços que possuem taxa de partida. Estes resultados mostram a força do algoritmo PD ao lidar com cenários onde decisões sobre ativação e desativação de poços devem ser tomadas. As regras *ad hoc* propostas em [1] requerem intervenção humana para decidir quais poços não surgentes serão ativados, e sendo assim podem resultar em alocações sub-ótimas. Estas regras *ad hoc* consistem em retirar do problema de otimização (desativar) poços para os quais foram alocados a mínima taxa de injeção possível, e então resolver o problema novamente para os poços restantes. Porém, se existem m poços para os quais foram alocados a taxa mínima, existem 2^m possibilidades de ativação/desativação, tornando esta decisão bastante difícil. Em nossa abordagem estas decisões são tomadas pelo próprio algoritmo, produzindo a solução ótima.

O desempenho do algoritmo PD foi também comparado ao do *software* GOAL, em um cenário

Tabela 3.8: Alocação de gás de elevação, cenário sem taxas de partida: comparação entre SQP [1] e PD

	Poços	q_p (B/D)	q_i^{max} (MSCF/D)	Ganho com PD (%)
SQP	5	537,21	84.950	-0,000
PD		537,21	84.950	
SQP	13	7.708,03	22.484	-0,0008
PD		7.707,97	22.484	
SQP	25	16.872,89	54.425	-0,009
PD		16.871,39	54.425	
SQP	37	21.212,79	72.497	-0,005
PD		21.211,67	72.497	

Tabela 3.9: Alocação de gás de elevação, cenário com taxas de partida: comparação entre SQP [1] e PD

	Poços	q_p (B/D)	q_i^{max} (MSCF/D)	Ganho com PD (%)
SQP	12	4.559,44	36.325	+68,36
PD		7.676,35	36.325	
SQP	24	7.877,72	57.138	+39,74
PD		11.008,63	57.138	
SQP	36	15.395,04	75.199	+20,24
PD		18.510,45	75.199	
SQP	48	23.944,50	110.085	+12,66
PD		26.975,18	110.085	
SQP	56	27.929,89	120.087	+10,54
PD		30.874,25	120.087	

de 16 poços contido em [36]. Foram realizados quatro experimentos, os dois primeiros sob condições de suprimento alto de gás de elevação, e os dois últimos sob condições de baixo suprimento. O objetivo é a maximização do lucro obtido com a venda dos hidrocarbonetos extraídos. A Tabela 3.10 contém os resultados destes experimentos. O desempenho do algoritmo PD é levemente inferior ao do GOAL para os cenários com grande disponibilidade de gás de elevação, enquanto sob condições de baixa disponibilidade o algoritmo PD supera o desempenho do GOAL. O ganho com PD foi muito menos pronunciado que o ganho mostrado na Tabela 3.9 pois naqueles cenários haviam poços com taxa de partida, o que aumenta a natureza combinatória do problema e faz com que o algoritmo PD proporcione melhores resultados. Em outras palavras, quanto maior o número de possibilidades de ativação/desativação de poços, maior a necessidade de tratar o problema de maneira formal.

Tabela 3.10: Resultados de alocação de gás de elevação: comparação entre GOAL [36] e PD

	Gás de elevação	Lucro (\$/D)	q_i^{max} (MSCF/D)	Ganho com PD (%)
GOAL	Suprimento	87.135,34	7,5	-0,003
PD	alto	87.133,06	7,5	
GOAL	Suprimento	80.842,29	5,0	-0,001
PD	alto	80.841,42	5,0	
GOAL	Suprimento	70.563,58	1,6	+1,257
PD	baixo	71.443,46	1,6	
GOAL	Suprimento	62.278,88	1,0	+0,998
PD	baixo	62.900,72	1,0	

3.4 Generalizações do Problema

Nesta seção, acrescentamos algumas características ao problema inicial $P(\emptyset)$ de modo a torná-lo mais geral e representativo.

Quando o nível de confiança das medições obtidas nos testes de produção é conhecido, podemos associar probabilidades aos coeficientes da função de aproximação, e a otimização de $P(\emptyset)$ seria realizada sobre os valores esperados destes coeficientes. Este problema pode ser formulado como:

$$\begin{aligned}
 P_E(\emptyset) : J_E = & \text{Maximizar} \quad \sum_{n=1}^N (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - \sum_{n=1}^N p_i q_i^n \\
 & \text{Sujeito a :} \\
 & \tilde{q}_p^n = f(q_i^n) \quad n = 1, \dots, N \\
 & \sum_{n=1}^N q_i^n \leq q_i^{max} \\
 & l_n y_n \leq q_i^n \leq u_n y_n \quad n = 1, \dots, N \\
 & y_n \in \{0, 1\} \quad n = 1, \dots, N
 \end{aligned} \tag{3.5}$$

onde os coeficientes da função \tilde{q}_p^n que representa o comportamento do poço n são valores esperados dos coeficientes das curvas definidos no Apêndice A.

Na ausência de dados sobre a confiabilidade dos testes, a otimização pode ser realizada levando em conta todas as $\kappa(n)$ curvas obtidas com os testes de produção disponíveis para o poço n , e fornecer as taxas de injeção de gás que maximizam o lucro no pior caso (baseado nas curvas de menor produ-

tividade), resultando no seguinte problema:

$$\begin{aligned}
 P_W(\emptyset) : J_W = & \text{ Maximizar } \sum_{n=1}^N \min\{(p_o\gamma_o^n + p_g\gamma_g^n - p_w\gamma_w^n)q_p^{n,k} - p_i q_i^n : k = 1, \dots, \kappa(n)\} \\
 & \text{ Sujeito a :} \\
 & q_p^{n,k} \in Q_n \quad n = 1, \dots, N, \quad k = 1, \dots, \kappa(n) \\
 & \sum_{n=1}^N q_i^n \leq q_i^{max} \\
 & l_n y_n \leq q_i^n \leq u_n y_n \quad n = 1, \dots, N \\
 & y_n \in \{0, 1\} \quad n = 1, \dots, N
 \end{aligned} \tag{3.6}$$

onde $Q_n = \{q_p^{n,k}(q_i^n) : k = 1, \dots, \kappa(n)\}$ é um conjunto de funções q_p^n obtidas em $\kappa(n)$ testes para o poço n .

Corolário 3.2 $P_W(\emptyset)$ é NP-difícil.

É fácil ver que um algoritmo que resolve $P_W(\emptyset)$ resolve também $P_E(\emptyset)$. Fornecendo $\kappa(n) = 1$ curvas de desempenho para todos os n poços (ou seja, uma WPC por poço) ao algoritmo que resolve $P_W(G)$, onde cada WPC representa o comportamento *esperado* do poço correspondente, o algoritmo produzirá uma solução para $P_E(G)$.

3.4.1 Solução de $P_W(\emptyset)$

A formulação de $P_W(\emptyset)$ é bastante similar à de $P(\emptyset)$, bastando considerar as curvas de menor produtividade no cálculo da função objetivo. Da mesma forma que em $P(\emptyset)$, é necessário discretizar a variável q_i^n para que o problema possa ser resolvido através de Programação Dinâmica. Sendo assim, o problema resultante P_{WM} é formulado da seguinte maneira:

$$\begin{aligned}
 P_{WM} : J_{WM} = & \text{ Maximizar } \sum_{n=1}^N \min\{(p_o\gamma_o^n + p_g\gamma_g^n - p_w\gamma_w^n)\tilde{q}_p^{n,k} - p_i q_i^n : k = 1, \dots, \kappa(n)\} \\
 & \text{ Sujeito a :} \\
 & \tilde{q}_p^{n,k} = f(q_i^n) \quad n = 1, \dots, N \\
 & \quad \quad \quad k = 1, \dots, \kappa(n) \\
 & l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \quad n = 1, \dots, N \\
 & \sum_{n=1}^N w_n \leq M \\
 & y_n \in \{0, 1\} \quad n = 1, \dots, N \\
 & w_n \in \{0, \dots, M\} \quad n = 1, \dots, N
 \end{aligned} \tag{3.7}$$

onde as curvas $\tilde{q}_p^{n,k}$ são aproximadas seguindo um dos três modelos apresentados no Apêndice A.

Limitando a taxa de injeção de gás disponível em $m \in \{0, \dots, M\}$ unidades padrão, as recorrências para a solução de P_{WM} via PD são dadas por:

$$\begin{aligned} \tilde{P}_W^{m,n} : F_W^{m,n} = & \text{Maximizar} \quad \min \{ (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^{k,n} - p_i q_i^n : k = 1, \dots, \kappa(n) \} \\ & \text{Sujeito a :} \\ & \tilde{q}_p^{n,k} = f(q_i^n) \quad k = 1, \dots, \kappa(n) \\ & l_n y_n \leq q_i^n \leq \min\{u_n, m\delta\} y_n \\ & y_n \in \{0, 1\} \end{aligned} \quad (3.8)$$

$$P_W^{m,N} : J_W^{m,N} = F_W^{m,N}$$

$$\begin{aligned} P_W^{m,n} : J_W^{m,n} = & \text{Maximizar} \quad \min \{ (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^{n,k} - p_i q_i^n : k = 1, \dots, \kappa(n) \} + J_W^{m-w_n, n+1} \\ & \text{Sujeito a :} \\ & \tilde{q}_p^{n,k} = f(q_i^n) \quad k = 1, \dots, \kappa(n) \\ & l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \\ & y_n \in \{0, 1\} \\ & w_n \in \{0, \dots, m\} \end{aligned} \quad (3.9)$$

Definidas estas recorrências, o algoritmo PD que resolve $P(\emptyset)$ apresentado na Seção 3.2.2 pode ser utilizado para resolver $P_W(\emptyset)$. A solução de P_{WM} é obtida em tempo $\Theta(KNM^2)$, onde $K = \max\{\kappa(n) : n = 1, \dots, N\}$ e demanda $\Theta(MN)$ unidades de memória para armazenar a tabela de valores $J_W^{m,n}$. Para verificar o desempenho do algoritmo quando o número de WPC s é maior que 1, foram realizados 36 experimentos, variando o número de poços ($N = 6, 12, 24$ e 48), taxa de injeção disponível (três valores de q_i^{max}), e a variância das WPC s em torno de uma WPC média (10%, 50% e 100%). O nível de discretização utilizado foi $M = 100$ e o número de curvas para todos os poços foi $\kappa(n) = 10$. Em média o algoritmo levou 47,42s para encontrar a solução, e ficou a 0,49% do limite superior calculado utilizando relaxação contínua e programação côncava [27].

3.4.2 Incertezas nas WPC s

Quando o nível de incerteza sobre as WPC s é muito alto, ou seja, quando o histórico de testes de um poço mostra WPC s muito diferentes, o engenheiro de produção de petróleo pode desejar assumir uma posição mais conservadora e basear a decisão sobre a quantidade de gás injetada em cada poço nas curvas que representam o pior caso (menor produtividade).

Apesar de esta abordagem ser intuitivamente menos lucrativa, existem casos onde o algoritmo de pior caso fornece melhores soluções do que as obtidas utilizando a média das curvas. Isto acontece

Tabela 3.11: Comparação entre desempenho do pior caso e caso médio

q_i^{max}	Teste 1		Teste 2	
	J_w/J_r	J_m/J_r	J_w/J_r	J_m/J_r
0,1	1,00	0,68	0,45	1,00
0,6	1,00	0,67	0,45	1,00
1,1	1,00	0,67	0,45	1,00
1,6	1,00	0,67	0,45	1,00
2,1	0,99	0,85	0,70	0,93
2,6	0,99	0,75	0,57	0,99
3,1	1,00	0,84	0,71	1,00
3,6	1,00	0,84	0,71	1,00
4,1	1,00	1,00	1,00	1,00
Média	1,00	0,77	0,61	0,99

quando é maior a probabilidade do comportamento real do poço ser representado pelas curvas de menor produtividade.

Foram realizados alguns testes numéricos para ilustrar situações onde cada caso possui melhor desempenho, de acordo com os seguintes passos: **1)** 10 *WPCs* foram fornecidas para cada poço; **2)** o algoritmo foi executado para $\kappa(n) = 10$, encontrando as taxas de injeção ótimas q_{iw} para o pior caso; **3)** o algoritmo foi executado para as *WPCs* médias de cada poço, ou seja, $\kappa(n) = 1$, encontrando as taxas de injeção ótimas q_{im} ; **4)** uma das 10 *WPCs* é tomada aleatoriamente para representar o comportamento real de cada poço, e o algoritmo é executado novamente para $\kappa(n) = 1$, obtendo a solução ótima “real” J_r ; **5)** utilizando q_{iw} , q_{im} , e as curvas que representam o comportamento real, encontramos os valores objetivo “reais” que seriam obtidos com as taxas de injeção obtidas para o pior caso (J_w) e para o caso médio (J_m); **6)** as relações J_w/J_r e J_m/J_r são comparadas; **7)** a taxa de injeção disponível q_i^{max} é variada, e os passos 2 a 6 são repetidos.

Os testes foram realizados para valores de q_i^{max} variando de 0,1 a 4,1, e o sorteio das curvas do passo 4 seguiram uma tendência para as curvas de pior desempenho no Teste 1, e uma tendência para o caso médio no Teste 2. Os resultados obtidos são mostrados na Tabela 3.11. Como esperado, o desempenho do pior caso é melhor no Teste 1, pois a probabilidade do comportamento real do poço ser representado pelas piores curvas é maior, e o desempenho do caso médio é melhor no Teste 2, pois a probabilidade do poço se comportar de acordo com as médias históricas das *WPCs* é maior. Esta probabilidade pode ser inferida pelo engenheiro de produção, baseada nos históricos de testes, cabendo a ele a decisão de quando fornecer uma ou múltiplas *WPCs* ao algoritmo. Nos experimentos aqui realizados, observou-se que quanto maior a taxa de gás disponível, mais os desempenhos se aproximam, pois o aspecto combinatório do problema torna-se menos relevante. No caso extremo, onde existe gás disponível para operar todos os poços ($q_i^{max} = 4,1$), os desempenhos se igualam.

3.5 Desigualdades válidas para $P(\emptyset)$

As similaridades entre $P(\emptyset)$ e o problema da mochila permite que as propriedades combinatórias deste último (em particular, famílias de desigualdades válidas que induzem facetes) sejam estendidas ao domínio de $P(\emptyset)$. A seguir, estendemos as desigualdades de cobertura para o problema da mochila e seu processo de *lifting* para $P(\emptyset)$ [31], [42].

Seja $X = \{(q_i, y) \in \mathbb{R}^N \times \mathbb{B}^N : (q_i, y) \text{ satisfaz as restrições de (3.1)}\}$ o conjunto de soluções factíveis para $P(\emptyset)$ e seja $\mathcal{P} = \text{conv}(X)$ o poliedro correspondente ao fecho convexo de X . Seja ainda $\mathcal{N} = \{1, \dots, N\}$ o conjunto de índices dos poços.

Proposição 3.2 *Se $l_n < u_n \leq q_i^{\max}$ para cada n , então $\dim(\mathcal{P}) = 2N$.*

Prova: Seja $s_n(\hat{q}_i^n, \hat{y}_n)$ o elemento de \mathcal{P} obtido quando $y_n = \hat{y}_n$, $y_k = 0$ para todo $k \neq n$, $q_i^n = \hat{q}_i^n$, e $q_i^k = 0$ para todo $k \neq n$. Seja $S = \{s_n(l_n, 1), s_n(u_n, 1) : n = 1, \dots, N\}$. Claramente, S é um conjunto de $2N$ elementos linearmente independentes de \mathcal{P} . Como $0 \in \mathcal{P}$, \mathcal{P} possui $2N + 1$ soluções afim independentes e, portanto, $\dim(\mathcal{P}) = 2N$. ■

No restante da seção, assumimos que as condições da Proposição 3.2 são satisfeitas.

Definição 3.1 *Um conjunto $C \subseteq \{1, \dots, N\}$ é uma cobertura se $\sum_{n \in C} l_n > q_i^{\max}$ e para cada $n \in C$, $\sum_{k \in C - \{n\}} l_k < q_i^{\max}$.*

Corolário 3.3 *O poliedro $\mathcal{P}_C = \{(q_i, y) \in \mathcal{P} : y_n = 0, \forall n \in \mathcal{N} - C\}$ induzido por uma cobertura C tem dimensão $\dim(\mathcal{P}_C) = 2|C|$.*

Proposição 3.3 *Se C é uma cobertura, então*

$$\sum_{n \in C} y_n \leq |C| - 1 \quad (3.10)$$

é uma desigualdade válida para \mathcal{P}_C . Além disso, a face $F_C = \{(q_i, y) \in \mathcal{P}_C : \sum_{n \in C} y_n = |C| - 1\}$ induzida pela desigualdade de cobertura possui dimensão $\dim(F_C) = 2|C| - 1$.

Prova: (Validade) Se $\sum_{n \in C} y_n = |C|$ para alguma solução $(q_i, y) \in \mathcal{P}_C$, então $\sum_{n \in C} q_i^n \geq \sum_{n \in C} l_n > q_i^{\max}$, o que significa que a solução é infactível, contradizendo a hipótese.

(Dimensionalidade) Seja $\sum_{n \in C} (\pi_n y_n + \mu_n q_i^n) \leq \pi_o$ uma desigualdade válida para \mathcal{P} induzindo uma face máxima $F_\pi = \{(q_i^n, y_n) \in \mathcal{P}_C : \sum_{n \in C} (\pi_n y_n + \mu_n q_i^n) = \pi_o\}$ de \mathcal{P}_C tal que $F_C \subseteq F_\pi$. O desenvolvimento a seguir mostra que a desigualdade que induz F_C difere da desigualdade que induz F_π por uma constante multiplicativa positiva.

Primeiro, mostramos que $\mu_n = 0$ para todo $n \in C$. Para cada $n \in C$ execute as seguintes operações: seja $(\hat{q}_i, \hat{y}) = (0, 0)$ um elemento de \mathcal{P}_C ; defina $C_n = C - \{j\}$ para algum $j \in C, j \neq n$; e faça $\hat{y}_k = 1$ e $\hat{q}_i^k = l_k$ para todo $k \in C_n$. Claramente a solução resultante $(\hat{q}_i, \hat{y}) \in F_C$. Seja $(\tilde{q}_i, \tilde{y}) = (\hat{q}_i, \hat{y})$ porém com $\tilde{q}_i^n = \min\{u_n, q_i^{max} - \sum_{k \in C_n - \{n\}} l_k\}$. Como $(\hat{q}_i, \hat{y}), (\tilde{q}_i, \tilde{y}) \in F_C$, as seguintes equações devem ser satisfeitas para que (\hat{q}_i, \hat{y}) e (\tilde{q}_i, \tilde{y}) pertençam a F_π :

$$\begin{aligned} \sum_{k \in C_n} \pi_k \hat{y}_k + \sum_{k \in C_n} \mu_k \hat{q}_i^k &= \pi_o \\ \sum_{k \in C_n} \pi_k \tilde{y}_k + \sum_{k \in C_n} \mu_k \tilde{q}_i^k &= \pi_o \end{aligned}$$

Subtraindo a primeira equação da segunda, obtemos $\mu_n(\tilde{q}_i^n - \hat{q}_i^n) = 0$. Como $\sum_{k \in C_n} l_k < q_i^{max}$ e $l_n < u_n$, $(\tilde{q}_i^n - \hat{q}_i^n) > 0$ e portanto $\mu_n = 0$. Aplicando este raciocínio para cada elemento de C , conclui-se que $\mu_n = 0$ para cada $n \in C$.

Agora, mostramos que $\pi_n = \pi_k$ para cada par $n, k \in C$. Seja $C_n = C - \{k\}$ e defina (\hat{q}_i, \hat{y}) como: $\hat{q}_i^j = l_j$ e $\hat{y}_j = 1$ para cada $j \in C_n$. Seja $C_k = C - \{n\}$ e (\tilde{q}_i, \tilde{y}) definido como: $\tilde{q}_i^j = l_j$ e $\tilde{y}_j = 1$ para cada $j \in C_k$. Claramente, $(\hat{q}_i, \hat{y}), (\tilde{q}_i, \tilde{y}) \in F_C$ e portanto as equações a seguir devem ser satisfeitas se $(\hat{q}_i, \hat{y}), (\tilde{q}_i, \tilde{y}) \in F_\pi$:

$$\begin{aligned} \sum_{j \in C_n} \pi_j \hat{y}_j &= \pi_o \\ \sum_{j \in C_k} \pi_j \tilde{y}_j &= \pi_o \end{aligned}$$

Subtraindo a segunda igualdade da primeira, obtemos $\pi_n \hat{y}_n - \pi_k \tilde{y}_k = 0 \Rightarrow \pi_n = \pi_k$. Segue da solução (\hat{q}_i, \hat{y}) que $\sum_{j \in C_n} \pi_j \hat{y}_j = \pi_o \Rightarrow \sum_{j \in C_n} \pi_j = \pi_o \Rightarrow (|C| - 1)\pi_n = \pi_o$. Fazendo $\pi_o = |C| - 1$, deduz-se que $\pi_j = 1$ para todo $j \in C$. Portanto, $\sum_{n \in C} y_n \leq |C| - 1$ induz uma face máxima F_C de \mathcal{P}_C , ou seja, $\dim(F_C) = 2|C| - 1$. ■

De acordo com a Proposição 3.3, a desigualdade válida induzida por uma cobertura C definiria uma faceta se a cobertura englobasse todos os poços, i.e., $C = \mathcal{N}$. Uma maneira de fortalecer a desigualdade de cobertura é aumentá-la com os poços cujos limites inferiores de injeção de gás são maiores ou iguais ao limite inferior de cada poço em C . Mais precisamente, a *cobertura estendida* $E(C) = C \cup \{n \in \mathcal{N} - C : l_n \geq l_k \text{ para todo } k \in C\}$ induz a *desigualdade de cobertura estendida*:

$$\sum_{n \in E(C)} y_n \leq |C| - 1 \quad (3.11)$$

que é válida para \mathcal{P} . A desigualdade de cobertura estendida (3.11) é válida pois qualquer elemento i de $E(C) - C$ pode substituir qualquer elemento j de C , ou seja, $\sum_{n \in (C - \{j\}) \cup \{i\}} l_n \geq \sum_{n \in C} l_n > q_i^{max}$. A desigualdade de cobertura estendida *domina* a desigualdade de cobertura pois qualquer ponto que viola (3.10) também viola (3.11), mas um ponto que viola (3.11) pode não violar (3.10).

Outra maneira de fortalecer desigualdades válidas é o princípio chamado de *lifting* [31], [42], onde uma desigualdade válida de $\mathcal{P}_C = \mathcal{P} \cap \{(q_i, y) \in \mathbb{R}^N \times \mathbb{B}^N : y_n = 0, n \in \mathcal{N} - C\}$ pode ser estendida para se tornar válida para \mathcal{P} . Iniciando com uma sequência ordenada $s = \langle j_1, \dots, j_L \rangle$ dos itens de $\mathcal{N} - C$, o procedimento de *lifting* adiciona à desigualdade, uma a uma, as variáveis correspondentes aos elementos de s . Para adicionar a variável do primeiro poço, por exemplo, o procedimento produz a desigualdade $\sum_{n \in C} y_n + \alpha_{j_1} y_{j_1} \leq |C| - 1$ que é válida para $\mathcal{P}_{C \cup \{j_1\}}$, onde $\alpha_{j_1} = |C| - 1 - \max\{\sum_{n \in C} y_n : (q_i, y) \in \mathcal{P}_{C \cup \{j_1\}}, y_{j_1} = 1\}$. A desigualdade resultante induz uma face $F_{C \cup \{j_1\}}$ de $\mathcal{P}_{C \cup \{j_1\}}$ tal que $\dim(F_{C \cup \{j_1\}}) \geq \dim(F_C) + 1$. Se a solução do problema que calcula α_{j_1} atinge seu máximo em $q_i^{j_1} > l_{j_1}$, então pode-se mostrar que $\dim(F_{C \cup \{j_1\}}) = \dim(F_C) + 2$. Os passos descritos acima podem ser aplicados sistematicamente aos elementos restantes de s até que a *desigualdade válida de lifting*:

$$\sum_{n \in C} y_n + \sum_{k=1}^L \alpha_{j_k} y_{j_k} \leq |C| - 1 \quad (3.12)$$

seja obtida. A desigualdade resultante é válida para \mathcal{P} e pode induzir uma de suas facetas sob as condições expostas acima.

Há uma maneira simples para calcular limites inferiores β_n para os fatores de *lifting* α_n dos elementos $n \in \mathcal{N} - C$. Para $k \in \{0, \dots, |C|\}$, seja $C(k) \subseteq C$ um subconjunto tal que $|C(k)| = k$ e $\min\{l_n : n \in C(k)\} \geq \max\{l_n : n \in C - C(k)\}$ se $k > 0$, em outras palavras, $C(k)$ é o subconjunto de C com os k elementos que requerem as taxas de injeção mais altas para sua operação. Seja também $q(C') = \sum_{n \in C'} l_n$ para qualquer $C' \subseteq C$. Para $n \in \mathcal{N} - C$, podemos definir o limite inferior β_n como:

$$\begin{aligned} \beta_n &= \text{Maximizar } k \in \{0, \dots, |C|\} \\ &\text{Sujeito a: } l_n \geq q(C(k)) \end{aligned}$$

Assim, substituindo os fatores de *lifting* α_n em (3.12) por seus limites inferiores β_n , obtemos a *desigualdade de cobertura fortalecida estendida*:

$$\sum_{n \in C} y_n + \sum_{n \in E(C) - C} \beta_n y_n \leq |C| - 1 \quad (3.13)$$

que também é válida para \mathcal{P} . Note que (3.13) pode ser mais forte que (3.11) porém mais fraca que (3.12).

3.6 Sumário

Neste capítulo, foram desenvolvidos algoritmos de Programação Dinâmica para a solução do problema de alocação ótima de gás de elevação para uma determinada taxa disponível de injeção de gás, considerando poços com taxa de partida e múltiplas curvas de desempenho de *gas-lift* (WPCs). O

desempenho destes algoritmos mostrou-se bastante satisfatório quando avaliado através de experimentos numéricos e comparações com exemplos da literatura, onde foi constatado que são capazes de fornecer soluções quase-ótimas em tempo relativamente baixo. Os resultados desta seção foram publicados em [11], [28], [9], [29].

Capítulo 4

Solução Recursiva de $P(G)$

Neste capítulo acrescentamos uma restrição ao problema estudado no capítulo anterior, e denominamos $P(G)$ o problema resultante. Suponhamos que por algum motivo técnico ou estratégico existam restrições de precedência de ativação/desativação entre os poços, e.g., se o poço i estiver ativo então um certo poço j também deve estar ativo. Estas restrições de precedência podem ser representadas por um grafo acíclico G , onde o conjunto dos vértices $V(G)$ representa os poços e o conjunto das arestas $E(G)$ representa as restrições de precedência. Quando este grafo forma uma floresta, chamamos o problema de otimização resultante de $P(F)$ (Seção 4.2), e quando G é um grafo acíclico geral, denotamos o problema por $P(G)$ (Seção 4.3).

4.1 Formulação do Problema

O problema $P_W(G)$ de alocação de uma taxa de injeção de gás de elevação disponível q_i^{max} com restrições de precedência de ativação e considerando a curva de menor produtividade no caso de múltiplas WPC s, pode ser escrito da seguinte maneira:

$$\begin{aligned} P_W(G) : J_W = & \text{ Maximizar } \sum_{n=1}^N \min \{ (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) q_p^{n,k} - p_i q_i^n : k = 1, \dots, \kappa(n) \} \\ & \text{ Sujeito a :} \\ & q_p^{n,k} \in Q_p \quad n = 1, \dots, N, \quad k = 1, \dots, \kappa(n) \\ & l_n y_n \leq q_i^n \leq u_n y_n \quad n = 1, \dots, N \\ & \sum_{n=1}^N q_i^n \leq q_i^{max} \\ & y_i \leq y_j \quad \forall (i, j) \in E(G) \\ & y_n \in \{0, 1\} \quad n = 1, \dots, N \end{aligned} \tag{4.1}$$

onde:

- $Q_p = \{q_p^{n,k}(q_i^n) : k = 1, \dots, \kappa(n)\}$ é um conjunto de funções q_p^n obtidas em $\kappa(n)$ testes para o poço n . Para cada poço n e teste k , a função $q_p^{n,k}$ é aproximada por uma função $\tilde{q}_p^{n,k}$ que assume uma das formas definidas no Apêndice A;
- $G = (V, E)$ é um grafo acíclico direcionado cujos vértices representam os poços ($V = \{1, \dots, N\}$) e cujas arestas representam restrições de precedência sobre a operação dos poços, ou seja, $(i, j) \in E$ significa que o poço j deve estar ativo se i estiver ativo e portanto j precede i .

Assim como no capítulo anterior, um algoritmo que resolve $P_W(G)$ resolve também $P_E(G)$, bastando fazer $\kappa(n) = 1$ para $n = 1, \dots, N$ e definir os coeficientes de $q_p^{n,k}$ como os valores esperados destes coeficientes. Além disso, as Hipóteses 3.1 e 3.2 continuam sendo válidas para esta formulação, bem como o Corolário 3.1.

A proposição a seguir afirma que o problema $P(G)$ é NP-difícil no sentido forte [16]. Isto significa que não existe algoritmo que seja capaz de encontrar uma solução ótima para este problema em tempo pseudo-polinomial, a não ser que a classe de problemas P seja igual à classe NP.

Proposição 4.1 $P_W(G)$ é NP-difícil no sentido forte.

Prova: Podemos mostrar que $P_W(G)$ é NP-difícil no sentido forte através de uma redução do problema da mochila parcialmente ordenado (*partially ordered knapsack problem*, [16], [17]). Uma instância I_K do problema da mochila parcialmente ordenado é dado por um conjunto finito U , uma ordem parcial \prec de U , para cada $u \in U$ um peso $s(u) \in \mathbb{Z}_+$ e um valor $v(u) \in \mathbb{Z}_+$, e valores inteiros positivos b e k .

Existe $U' \subseteq U$ tal que se $u \in U'$ e $u' \prec u$, então $u' \in U'$, e tal que $\sum_{u \in U'} s(u) \leq b$ e $\sum_{u \in U'} v(u) \geq k$?

A resposta algorítmica a esta questão constitui o problema da mochila parcialmente ordenado. Uma instância I_K pode ser reduzida a uma instância I_P de $P_W(G)$ da seguinte maneira. Assuma, sem perda de generalidade, que $U = \{1, \dots, N\}$. Seja $q_i^{max} = b$, e para cada $n \in U$ faça $\kappa(n) = 1$, $l_n = u_n = s(u)$, $q_p^n = y_u v(u)$ (para a função polinomial, basta fazer $\alpha_0^n = v(u)$ e $\alpha_1^n = \alpha_2^n = \alpha_3^n = 0$; para a função exponencial, $A_n = v(u)$ e $B_n = C_n = D_n = 0$; para a função logarítmica, $c_1 = v(u)$ e $c_2 = c_3 = c_4 = 0$). Defina $\gamma_o^n = 1$ e $\gamma_g^n = \gamma_w^n = 0$ para cada n , e faça $p_o = 1$, $p_g = p_w = p_i = 0$. Seja $G = (V, E)$ onde $V = U$ e $(u, u') \in E$ se e somente se $u' \prec u$. Então uma solução S_P para I_P tem valor objetivo maior ou igual a k se e somente se existir $U' \subseteq U$ com as propriedades desejadas, em outras palavras, $u \in U'$ se e somente se $y_u = 1$. ■

A seguir apresentamos o algoritmo PD para $P_E(F)$ e o algoritmo recursivo para $P_E(G)$, bem como resultados numéricos que comprovam a eficiência destes algoritmos. Optamos por resolver a versão P_E do problema por ser mais simples que P_W , e no final do capítulo indicamos como podem ser resolvidos os problemas $P_W(F)$ e $P_W(G)$. As próximas seções utilizam a seguinte notação ao referir-se a um grafo G :

- $\delta^+(u) = \{v : (u, v) \in E(G)\}$ é o conjunto de vértices diretamente acessíveis por u ;
- $\delta^-(u) = \{v : (v, u) \in E(G)\}$ é o conjunto de vértices que podem alcançar u diretamente;
- $d^+(u) = |\delta^+(u)|$ é o grau de saída do vértice u ;
- $d^-(u) = |\delta^-(u)|$ é o grau de entrada de u ;
- $S_G(u) = \{v : \text{existe um caminho em } G \text{ de } u \text{ para } v\}$ é o conjunto de descendentes de u no grafo G ;
- $S_G^+(u) = S_G(u) - \{u\}$;
- $\text{root}(G) = \{u : d^-(u) = 0\}$ é o conjunto das raízes de G .

4.2 Solução de $P(F)$

Nesta seção, além da taxa disponível para injeção de gás de elevação, os poços em $P_E(F)$ estão acoplados por restrições de ativação impostas pelas arestas de uma floresta F . Para decompor $P_E(F)$ de forma ótima em uma série de problemas côncavos desacoplados, um para cada poço, seria preciso saber a distribuição ótima das taxas de injeção de gás que satisfazem as restrições de precedência. Nossa abordagem lida primeiro com a questão da distribuição de gás, discretizando a taxa disponível em M unidades padrão, $\delta = q_i^{\max}/M$, e então trata as restrições de precedência através da ordem topológica induzida pela floresta F . Da mesma forma que no cenário sem restrições de precedência, a enumeração explícita de todas as alocações factíveis pode ser impraticável. Em particular, uma chamada para o algoritmo $C_\#(F)$ definido abaixo computa o número de ativações de poços que respeitam as restrições induzidas por um grafo do tipo floresta F em tempo linear¹.

Algoritmo $C_\#(F)$

-
- 1) Se $|V(F)| = 1$ então
 - 2) retorne 2
 - 3) Se $|\text{root}(F)| = 1$ então
 - 4) retorne $1 + C_\#(F[V(F) - \text{root}(F)])$
 - 5) $w \leftarrow 1$
 - 6) Para cada $u \in \text{root}(F)$ faça
 - 7) $w \leftarrow w * C_\#(F[S_F(u)])$
 - 8) retorne w
-

¹O algoritmo $C_\#(F)$ pode ser facilmente aumentado para enumerar todas as ativações factíveis.

Proposição 4.2 $C_{\#}(F)$ calcula corretamente o número de ativações de poços que atende às restrições impostas por um grafo direcionado do tipo floresta F .

Prova: (Por indução na cardinalidade n de $V(F)$). Para $n = 1$ o número reportado por $C_{\#}$ é claramente correto. Para $n \geq 2$, dois casos devem ser considerados: (*caso 1*) se $|root(F)| = 1$, então $C_{\#}(F)$ computa a ativação da raiz de F somada a todas as ativações factíveis de $F[V(F) - root(F)]$ quando a raiz não está ativa, o que é corretamente calculado pela hipótese de indução; (*caso 2*) se $|root(F)| = m \geq 2$, então seja $F = T_1 \cup \dots \cup T_m$, onde T_j , $j = 1, \dots, m$, são as sub-árvores com raízes nos vértices que aparecem em $root(F)$; como os poços que estão em uma sub-árvore são independentes daqueles que estão em outras sub-árvores, segue por hipótese de indução que o número de ativações é $\prod_{j=1}^m C_{\#}(T_j)$, que é precisamente o que o algoritmo $C_{\#}$ calcula. ■

Uma versão não recursiva do algoritmo $C_{\#}(F)$ pode ser implementada para computar o número de ativações de F com tempo de execução $\Theta(n)$ onde $n = |V|$. Para tanto, basta produzir uma sequência $\langle k_1, \dots, k_n \rangle$ dos vértices de F em ordem topológica reversa, ou seja, para todo $k_j \in V$ e todo $k_i \in S_F^+(k_j)$ tem-se $i < j$. Depois, calcula-se $C_{\#}(k_i)$ (número de ativações factíveis para a sub-árvore de F com raiz em k_i): para $i = 1, \dots, n$ faça $C_{\#}(k_i) = 2$ se k_i é uma folha, caso contrário faça $C_{\#}(k_i) = 1 + \prod_{j \in S_F^+(k_i)} C_{\#}(j)$. Por fim, $C_{\#}(F) = \prod_{j \in root(F)} C_{\#}(j)$.

4.2.1 Recursões

O problema $P_E(F)$ pode ser aproximado através da alocação de valores discretos de unidades padrão de injeção de gás. No restante desta seção, P é uma abreviação para $P_E(F)$ e P_M denota a versão discretizada de P , onde $\delta = q_i^{max}/M$ representa uma unidade padrão de alocação de gás e M é o número de unidades de injeção disponível. O problema P_M é definido como:

$$\begin{aligned}
 P_M : J_M = \quad & \text{Maximizar} \quad \sum_{n=1}^N (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - \sum_{n=1}^N p_i q_i^n \\
 & \text{Sujeito a :} \\
 & \quad \tilde{q}_p^n = f(q_i^n) \quad n = 1, \dots, N \\
 & \quad l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \quad n = 1, \dots, N \\
 & \quad \sum_{n=1}^N w_n \leq M \\
 & \quad y_i \leq y_j \quad \forall (i, j) \in E(F) \\
 & \quad y_n \in \{0, 1\} \quad n = 1, \dots, N \\
 & \quad w_n \in \{0, \dots, M\} \quad n = 1, \dots, N
 \end{aligned} \tag{4.2}$$

onde a função \tilde{q}_p^n é a função que aproxima o comportamento da produção de fluidos do poço n em função de q_i^n , utilizando um dos modelos de aproximação mostrados no Apêndice A, e w_n é o número de unidades padrão de taxa de injeção de gás alocado para o poço n .

Corolário 4.1 $P_M(G)$ é NP-difícil no sentido forte.

O algoritmo PD que será apresentado a seguir faz uso das soluções dos seguintes problemas recursivos:

- $P_{m,R}^y$ é a versão discretizada de P restrita aos poços que são descendentes dos elementos de $R \subseteq V(F)$ da floresta F , quando a taxa de injeção disponível for de m unidades padrão, e tal que as árvores que possuem raízes nos vértices de R são disjuntas; cada poço $r \in R$ deve estar ativo se $y = 1$, caso contrário o poço pode ou não estar ativo, ou seja $y_r \geq y$ para todo $r \in R$, e a taxa disponível de injeção de gás é de m unidades padrão para todos os elementos de R e seus descendentes, $\cup_{r \in R} S_F(r)$; $J_{m,R}^y$ é o valor ótimo da função objetivo de $P_{m,R}^y$.
- $P_{m,n}^{y_n}$ é a versão discretizada de P restrita ao poço n e seus descendentes em F , $S_F^+(n)$, onde a ativação do poço n é determinada por y_n , e existem m unidades padrão de taxa de injeção de gás para todos os poços em $S_F(n)$; $J_{m,n}^{y_n}$ é o valor ótimo da função objetivo para $P_{m,n}^{y_n}$, podendo assumir valor $-\infty$ caso $m\delta$ não seja suficiente para ativar n e seus descendentes em F .

De acordo com a terminologia acima, a solução ótima de P_M é a solução do problema $P_{M,root(F)}^0$ ($J_M = J_{M,root(F)}^0$), que pode ser resolvido através das seguintes recursões:

$$\begin{aligned}
 P_{m,n}^{y_n} : J_{m,n}^{y_n} = & \text{ Maximizar } (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n + J_{m-w_n, \delta^+(n)}^{y_n} \\
 & \text{ Sujeito a :} \\
 & \tilde{q}_p^n = f(q_i^n) \\
 & l_n y_n \leq q_i^n \leq \min\{u_n, w_n \delta\} y_n \\
 & w_n \in \{0, \dots, m\} \\
 & y_n \in \{0, 1\}
 \end{aligned} \tag{4.3}$$

$$\begin{aligned}
 P_{m,R}^y : J_{m,R}^y = & 0 \quad \text{se } R = \emptyset \\
 J_{m,R}^y = & \text{ Maximizar } \max\{J_{w_p,p}^k : k \geq y, k \in \{0, 1\}\} + J_{m-w_p, R-\{p\}}^y \\
 & \text{ Sujeito a :} \\
 & p \in R \\
 & w_p \in \{0, \dots, m\}
 \end{aligned} \tag{4.4}$$

4.2.2 Algoritmo PD para $P(F)$

Para resolver as recursões (4.3)-(4.4), a ordem topológica imposta pelas arestas de F é utilizada, ou seja, o conjunto de subproblemas $\{P_{m,n}^{y_n}\}$ pode ser resolvido a partir dos poços que compõem as folhas de F , em direção às raízes de F , respeitando a ordem topológica.

Algoritmo PD para P_M **Inicialização**

Seja $\Lambda = (\lambda_1, \dots, \lambda_N)$ uma permutação de $\{1, \dots, N\}$ obedecendo uma ordem topológica de F .

Recorrência

Para $k = N, \dots, 1$:

$n \leftarrow \lambda_k$

Se $d^+(n) = 0$ então:

Para $m = 0, \dots, M$:

Resolva $P_{m,n}^{y_n}$ para $y_n \in \{0, 1\}$, obtendo $J_{m,n}^{y_n}$

Senão:

Seja $\{r_1, \dots, r_d\} = \delta^+(n)$, onde $d = d^+(n)$

Seja $J_{m,\{r_d\}}^1 \leftarrow J_{m,r_d}^1$ para $m = 0, \dots, M$

Para $t = d - 1$ até 1:

Para $m = 0, \dots, M$:

Para $w_{r_t} = 0, \dots, m$:

Resolva $P_{m,\{r_t, \dots, r_d\}}^1$ com $p = r_t$ para obter $J_{m,\{r_t, \dots, r_d\}}^1$

Seja $J_{m,\{r_d\}}^0 \leftarrow \max\{J_{m,r_d}^0, J_{m,r_d}^1\}$ para $m = 0, \dots, M$

Para $t = d - 1$ até 1:

Para $m = 0, \dots, M$:

Para $w_{r_t} = 0, \dots, m$:

Resolva $P_{m,\{r_t, \dots, r_d\}}^0$ com $p = r_t$ para obter $J_{m,\{r_t, \dots, r_d\}}^0$

Para $m = 0, \dots, M$:

Para $w_n = 0, \dots, m$:

Resolva $P_{m,n}^{y_n}$ para obter $J_{m,n}^{y_n}$ para $y_n \in \{0, 1\}$

Fim

Finalização

Seguindo os passos do laço *Senão* acima, substituindo o conjunto $\{r_1, \dots, r_d\}$ por $root(F)$, onde $d = |root(F)|$, resolva $P_{m,root(F)}^y$ para obter $J_{m,root(F)}^y$ para $y \in \{0, 1\}$ e $m = \{0, \dots, M\}$.

A Figura 4.1 ilustra como o algoritmo resolve as recursões para um grupo de 6 poços e um grafo de restrições de precedência simples. Para facilitar o entendimento da figura, denotamos o conjunto de sub-problemas $\{P_{m,n}^{y_n} : m = 0, \dots, M\}$ de $\mathcal{P}_n^{y_n}$ e, similarmente, o conjunto $\{P_{m,R}^y : m = 0, \dots, M\}$ de \mathcal{P}_R^y . Estendendo a notação, temos $\mathcal{J}_n^{y_n} = \{J_{m,n}^{y_n} : m = 0, \dots, M\}$ e $\mathcal{J}_R^y = \{J_{m,R}^y : m = 0, \dots, M\}$.

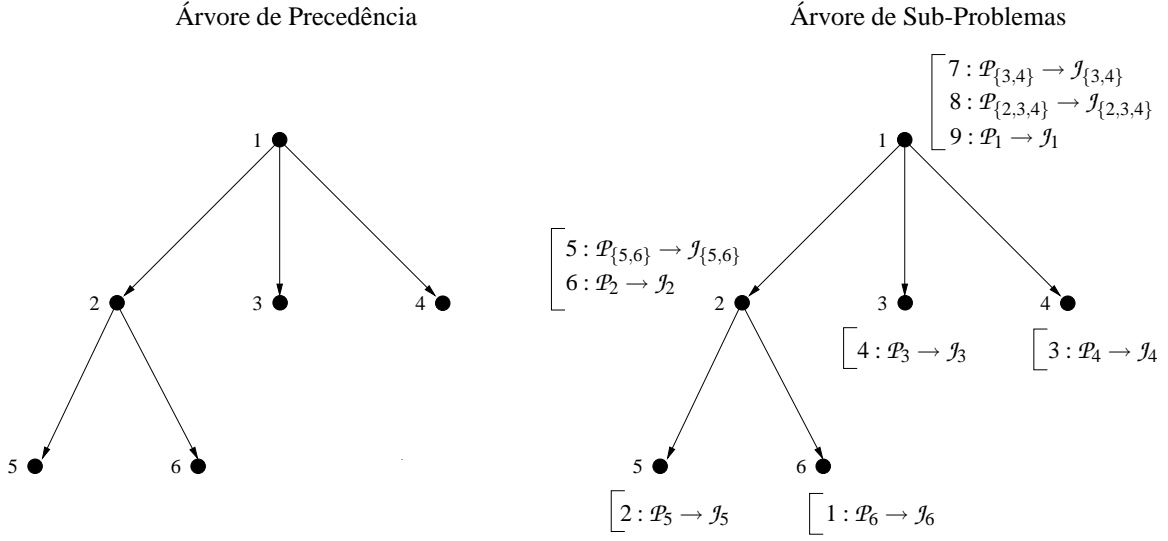


Figura 4.1: Ilustração do algoritmo PD para $P(F)$. O grafo da esquerda define as restrições de precedência de ativação dos poços. Para uma ordem topológica $T = \langle 1, 2, 3, 4, 5, 6 \rangle$, os conjuntos de sub-problemas resolvidos e as tabelas geradas pelo algoritmo são mostrados na árvore da direita, que indica a ordem na qual os problemas são resolvidos. Por exemplo, $5 : \mathcal{P}_{\{5,6\}} \rightarrow \mathcal{J}_{\{5,6\}}$ significa que o conjunto de sub-problemas $\mathcal{P}_{\{5,6\}}^y$ foi o quinto a ser resolvido para obter $\mathcal{J}_{\{5,6\}}^y$, para $y \in \{0, 1\}$.

Dois questões relevantes que surgem na análise deste algoritmo dizem respeito ao número de tabelas que o algoritmo necessita armazenar, e ao número de passos computacionais necessários para o preenchimento das mesmas.

Considerando o número de tabelas, o algoritmo computará $2N$ tabelas $\mathcal{J}_n^{y_n}$ (uma para cada n e y_n) e $2 \sum_{n=1}^N d^+(n)$ tabelas \mathcal{J}_R^y , duas para cada $R_t^n = \{r_1, \dots, r_d\} \subseteq \{r_1, \dots, r_d\} = \delta^+(n)$, $t = 1, \dots, d = d^+(n)$, além de $2|\text{root}(F)|$ tabelas \mathcal{J}_R^y onde $R \subseteq \text{root}(F)$. Como cada tabela possui $\Theta(M)$ elementos, juntas elas ocuparão $\Theta(2M(N + \sum_{n=1}^N d^+(n) + |\text{root}(F)|)) \in \Theta(4MN)$ unidades de memória.

Sobre o número de passos computacionais, dividimos a solução dos subproblemas entre folhas e não-folhas de F . Para um poço folha n , o custo para obter $\mathcal{J}_n^{y_n}$ é da ordem de $\Theta(M)$. O custo computacional associado a um poço não-folha por sua vez é dividido entre o cálculo das tabelas \mathcal{J}_R^y e \mathcal{J}_n^y . O primeiro é equivalente a resolver um problema $P_M(0)$ com $d^+(n)$ “super-poços”², onde as funções objetivo dos “super-poços” são induzidas pelos valores das tabelas já computadas, \mathcal{J}_k^y para $k \in \delta^+(n)$; portanto, o cálculo de \mathcal{J}_R^y termina em $\Theta(d^+(n)M^2)$ passos. O segundo cálculo, \mathcal{J}_n^y pode ser realizado em $\Theta(M^2)$ passos. No total, conclui-se que o tempo de execução do algoritmo é da ordem de $\Theta(4NM^2)$.

²Um “super-poço” consiste de uma sub-árvore de poços com raiz em um filho particular do vértice n .

Tabela 4.1: Soluções de $P(F)$ para uma instância de 6 poços, com q_i^{max} variável e $M = 200$.

q_i^{max}	q_i^1	q_i^2	q_i^3	q_i^4	q_i^5	q_i^6	J
50	7.4251	7.6954	7.4406	7.2722	7.0173	7.0379	989.1743
40	6.8000	7.2000	6.8000	6.6000	6.2000	6.4000	977.9290
30	5.1000	5.7000	5.2500	4.6500	4.3500	4.9500	858.0924
20	0.0000	4.7000	4.2000	3.7000	3.7000	3.7000	609.6331
15	0.0000	3.9750	3.6750	0.0000	3.6750	3.6750	476.0638
13	0.0000	5.1350	0.0000	0.0000	3.7050	4.1600	396.1391
10	0.0000	0.0000	5.2000	0.0000	0.0000	4.8000	294.9611
7	0.0000	0.0000	7.0000	0.0000	0.0000	0.0000	177.9975

Para ilustrar o funcionamento do algoritmo, utilizamos a mesma instância de 6 poços apresentada na Seção 3.2.2, e o grafo de restrições de precedência da Figura 4.1. A Tabela 4.1 mostra como a solução varia para diversas taxas de injeção disponíveis q_i^{max} e um nível de discretização $M = 200$. Para $q_i^{max} = 50$, existe gás suficiente para ativar todos os poços em seu ponto ótimo. É interessante observar como os poços são desativados com a diminuição da taxa de injeção disponível. Note que, por exemplo, o poço 3 é desativado quando a taxa disponível cai de 15 para 13, e volta a ser ativado quando a taxa cai para 10. Isso ilustra como a natureza combinatória do problema pode fazer com que as técnicas *ad hoc* normalmente utilizadas forneçam soluções sub-ótimas.

4.2.3 Resultados Numéricos

O algoritmo apresentado foi implementado em sistema operacional GNU Linux e linguagens de programação C/C++, e então testado em uma série de 24 experimentos numéricos, variando o número de poços, o nível de discretização e a estrutura das restrições de precedência. A Tabela 4.2 mostra os resultados destes experimentos, contendo a razão J_M/UB e o tempo de execução do algoritmo. O valor $J_M = J_{M,root(F)}^0$ representa a solução ótima encontrada pelo algoritmo PD, e UB representa um limite superior para J_M , obtido através de um programa de otimização convexa [27].

Os experimentos mostram que apesar de o algoritmo PD fornecer uma solução aproximada, esta é bastante próxima do limite superior teórico, ou seja, o algoritmo PD para P_M fornece soluções bastante próximas das soluções ótimas em um período de tempo razoável, considerando as instâncias utilizadas neste trabalho.

Tabela 4.2: Qualidade de aproximação e tempo de execução para $P_M(F)$

Número de poços	M	J_M/UB (%)	Tempo de execução (s)
6	10	96,65	0,06
	80	99,89	0,11
	320	99,94	6,35
6	10	94,72	0,93
	80	99,08	0,93
	320	99,08	5,28
12	20	94,82	0,03
	80	99,92	0,08
	160	99,97	5,16
12	20	96,27	1,97
	80	97,81	2,19
	320	98,03	3,73
24	50	94,14	3,52
	200	95,67	4,33
	800	96,48	5,86
24	50	97,70	0,70
	200	99,21	0,79
	800	99,30	2,31
36	100	94,18	3,20
	400	96,59	3,41
	800	96,79	5,82
36	100	94,97	2,26
	400	97,10	2,90
	800	97,74	5,03
Média		97,21	2,79

4.3 Solução de $P(G)$

Esta seção trata da versão onde as restrições de precedência de ativação são representadas por um grafo acíclico G . A solução do problema resultante segue a mesma linha dos problemas anteriores. Ao invés de resolvermos $P(G)$, resolvemos sua versão discretizada, $P_M(G)$, onde a taxa de injeção de gás disponível é dividida em M unidades padrão $\delta = q_i^{max}/M$, permitindo uma solução recursiva. O algoritmo “quebra” $P_M(G)$ em subproblemas que são resolvidos recursivamente e cujas soluções são armazenadas em uma estrutura de dados, posteriormente sendo integradas para obter a solução do problema original.

4.3.1 Recursões

Antes de formular as recursões, descrevemos brevemente os problemas que são resolvidos recursivamente pelo algoritmo e as tabelas produzidas. As recursões resolvem dois subproblemas:

- $P_m(H)$ é a versão do problema original restrita aos poços contidos no conjunto $V(H)$, as restrições de precedência representadas por $E(H)$, e uma quantidade de m unidades padrão de taxa de injeção de gás, onde H é um subgrafo de G ; $J_m(H)$ denota o valor ótimo da função objetivo para $P_m(H)$, onde $\mathcal{J}(H) = \{J_m(H) : m = 0, \dots, M\}$ é a tabela com os valores objetivo para todos os níveis de taxa de injeção de gás disponível; $\mathcal{P}(H)$ denota o conjunto de sub-problemas $\{P_m(H) : m = 0, \dots, M\}$.
- $P_m^r(H)$ é uma versão mais restrita de $P_m(H)$ onde o poço $r \in \text{root}(H)$ deve estar ativo; $J_m^r(H)$ representa o valor ótimo da função objetivo, que pode assumir valor $-\infty$ se a quantidade $m\delta$ não é suficiente para ativar r e seus descendentes em H ; $\mathcal{J}^r(H) = \{J_m^r(H) : m = 0, \dots, M\}$ é a tabela com os valores objetivo para todos os níveis de taxa de injeção de gás disponível, enquanto $\mathcal{P}^r(H) = \{P_m^r(H)\}$ denota o conjunto de sub-problemas correspondentes.

Seguindo a notação acima, pode-se estabelecer as recursões:

$$\begin{aligned}
 P_m(H) : \quad & \text{se } E(H) = \emptyset : \\
 & J_m(H) = \text{Maximizar} \quad \sum_{n \in V(H)} [(p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n] \\
 & \text{Sujeito a:} \\
 & \quad \tilde{q}_p^n = f(q_i^n), \forall n \in V(H) \\
 & \quad w_n \in \{0, \dots, m\}, \forall n \in V(H) \\
 & \quad \sum_{n \in V(H)} w_n = m \\
 & \quad l_n y_n \leq q_i^n \leq y_n \min\{u_n, \delta w_n\}, \forall n \in V(H) \\
 & \quad y_n \in \{0, 1\}, \forall n \in V(H) \\
 & \text{se } E(H) \neq \emptyset : \\
 & \quad J_m(H) = \max\{J_m^r(H), J_m(H[V(H) - \{r\}])\}, r \in \text{root}(H)
 \end{aligned} \tag{4.5}$$

$$\begin{aligned}
P_m^r(H) : \quad & \text{se } S_H(r) = V(H) : \\
& J_m^r(H) = \text{Maximizar} \quad \sum_{n \in V(H)} [(p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) \tilde{q}_p^n - p_i q_i^n] \\
& \text{Sujeito a:} \\
& \quad \tilde{q}_p^n = f(q_i^n), \quad \forall n \in V(H) \\
& \quad w_n \in \{0, \dots, m\}, \quad \forall n \in V(H) \\
& \quad \sum_{n \in V(H)} w_n = m \\
& \quad l_n \leq q_i^n \leq \min\{u_n, \delta w_n\}, \quad \forall n \in V(H) \\
& \text{se } S_H(r) \neq V(H) : \\
& J_m^r(H) = \text{Max} \quad J_k^r(H[S_H(r)]) + J_{m-k}(H[V(H) - S_H(r)]) \\
& \text{S. a:} \\
& \quad k \in \{0, \dots, m\}
\end{aligned} \tag{4.6}$$

Os algoritmos que resolvem as recursões acima são dados na próxima seção e utilizam duas estruturas de dados globais: um dicionário D [12] para armazenar as tabelas $J(H)$ e $J^r(H)$, e uma ordem topológica $T = \langle t_1, \dots, t_N \rangle$ de G para guiar a solução recursiva de $P_M(G)$.

4.3.2 Algoritmo Recursivo para $P(G)$

Os algoritmos $SolveJ(H)$ e $SolveJ(H, r)$, que respectivamente calculam recursivamente as tabelas $J(H)$ e $J^r(H)$ são descritos abaixo. Para resolver $P_M(G)$, deve-se obter uma ordem topológica $T = \langle t_1, \dots, t_N \rangle$ dos vértices de G e inicializar a estrutura de dados global e fazer uma chamada para $SolveJ(G)$. Ao término da execução deste algoritmo, a solução para $P_M(G)$ pode ser obtida a partir da tabela $J(G)$ e das tabelas armazenadas em D .

Algoritmo $SolveJ(H)$

Se $J(H) \in D$ então:

Retorne $J(H)$

Fim - Se

Se $E(H) = \emptyset$ então:

Resolva $P_m(H)$ para $m = 0, \dots, M$ e produza $J(H)$

Armazene $J(H)$ em D

Retorne $J(H)$

Senão:

$r \leftarrow t_k$, onde k é o menor índice tal que $t_k \in V(H)$ (Note que $r \in \text{root}(H)$)

$J^r(H) \leftarrow SolveJ(H, r)$

$J(H_r) \leftarrow SolveJ(H_r)$ onde $H_r = H[V(H) - \{r\}]$

Resolva $P_m(H)$ para $m = 0, \dots, M$ e produza $J(H)$

Armazene $J(H)$ em D

Retorne $J(H)$

Fim - Se

Algoritmo $SolveJ(H, r)$

Se $J^r(H) \in D$ então:

Retorne $J^r(H)$

Fim - Se

Se $V(H) = S_H(r)$ então:

Resolva $P_m^r(H)$ para $m = 0, \dots, M$ e obtenha $J^r(H)$

Armazene $J^r(H)$ em D

Retorne $J^r(H)$

Senão:

Seja $H_1 = H[S_H(r)]$;

$J^r(H_1) \leftarrow SolveJ(H_1, r)$

Seja $H_2 = H[V(H) - S_H(r)]$

$J(H_2) \leftarrow SolveJ(H_2)$

Resolva $P_m^r(H)$ para $m = 0, \dots, M$ e produza $J^r(H)$

Armazene $J^r(H)$ em D

Retorne $J^r(H)$

Fim - Se

A maneira como os algoritmos acima operam é ilustrada na Figura 4.2 para uma instância simples de 6 poços. O grafo com as restrições de precedência de ativação dos poços é mostrado na raiz da árvore de sub-problemas. O algoritmo segue a ordem topológica $T = \langle 3, 1, 2, 4, 5 \rangle$ para decompor os problemas em sub-problemas.

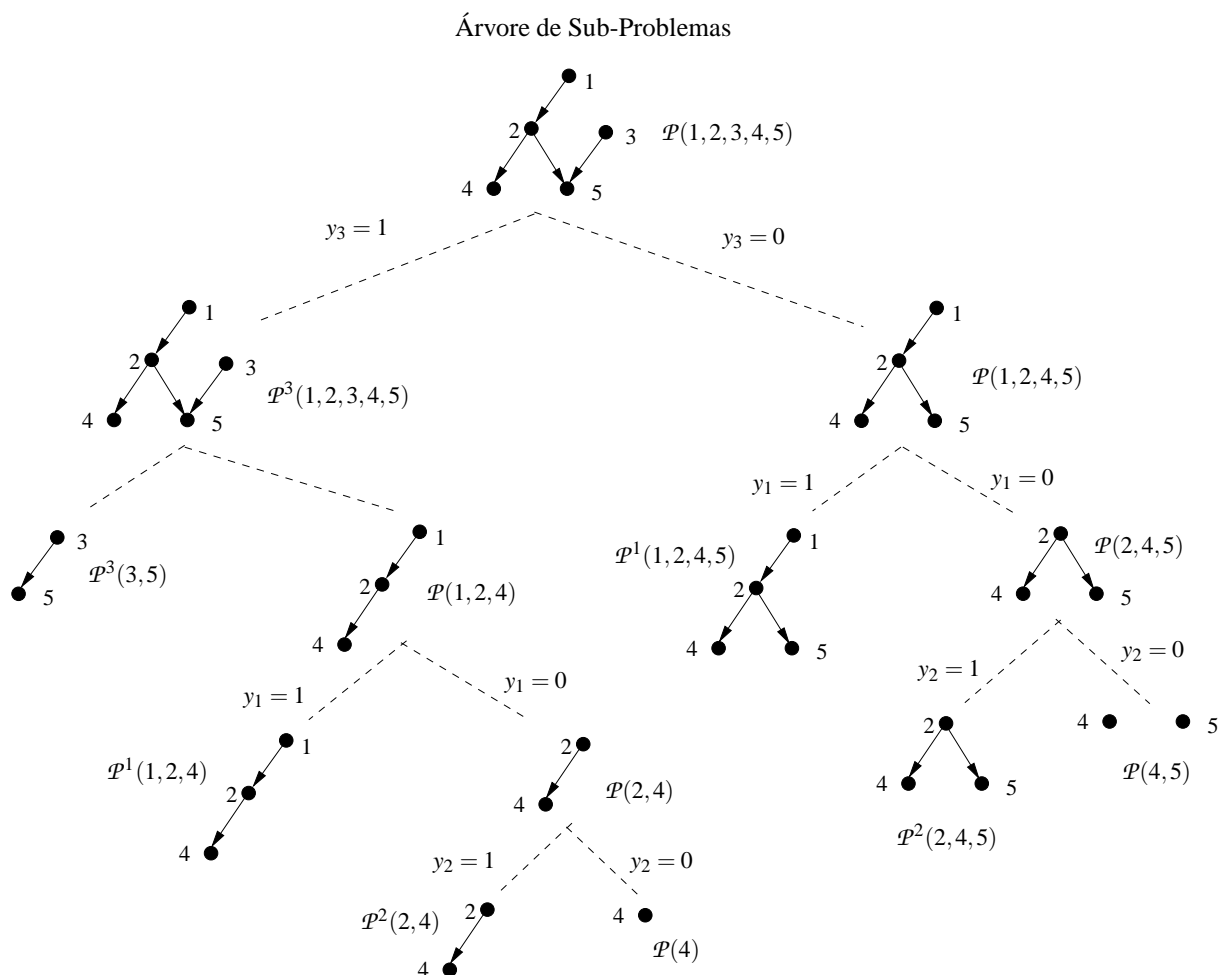


Figura 4.2: Ilustração dos algoritmos recursivos que resolvem $P(G)$. O grafo de restrições de precedência G possui 5 nós como ilustrado na raiz da árvore de sub-problemas. A ordem topológica de G usada para guiar a solução é $T = \langle 3, 1, 2, 4, 5 \rangle$. Cada nó indica um conjunto de sub-problemas resolvido pelo algoritmo: $P(1, 2, 4)$ representa o conjunto de sub-problemas $P(G[\{1, 2, 4\}])$.

Devido à necessidade de armazenarmos as tabelas já computadas³ e da solução de um problema equivalente a $P(\emptyset)$, quando $E(H) = \emptyset$, conclui-se que o tempo de execução do algoritmo pertence a $O(|D|NM^2)$, onde $|D|$ é o número de tabelas armazenadas em D .

Uma chamada de $SolveJ(G)$ pode levar tempo exponencial em algumas instâncias. Seja G um grafo representando restrições de precedência de ativação onde $V = \{1, \dots, 2k+1\}$ e $E = \{(1, k+2), (2, k+3), \dots, (k, 2k+1)\} \cup \{(k+1, k+2), \dots, (k+1, 2k+1)\}$. Para a ordem topológica $T = \langle 1, 2, \dots, 2k+1 \rangle$, $SolveJ(G)$ calcula $\Omega(2^k)$ tabelas distintas e consequentemente $|D| \in \Omega(2^k)$. Por outro lado, o tempo de execução pode ser linear no número de poços se a ordem topológica iniciar em $k+1$, o que ilustra que o desempenho do algoritmo depende tanto de G como de sua ordem topológica.

Apesar do desempenho desfavorável no pior caso, os tempos de execução do algoritmo para $P_M(G)$ são comparáveis aos tempos de execução do algoritmo para $P_M(F)$, para as instâncias consideradas neste trabalho. Os resultados dos experimentos numéricos que corroboram esta afirmação são mostrados na próxima seção.

4.3.3 Resultados Numéricos

O algoritmo proposto foi implementado utilizando linguagens de programação C/C++ e estruturas de dados (listas e dicionários) da biblioteca LEDA [25], em ambiente GNU Linux.

A qualidade com que $P_M(G)$ aproxima $P(G)$, bem como seu desempenho em termos de tempo computacional foram testados em 24 experimentos, variando o número de poços e a estrutura das restrições de precedência de ativação representadas pelo grafo G . A Tabela 4.3 contém os resultados destes experimentos, incluindo o número de poços, o tempo de execução do algoritmo, e a razão J_M/UB , sendo J_M o valor objetivo encontrado pelo algoritmo PD e UB um limite superior calculado através da relaxação das variáveis inteiras e programação côncava (*MINOS solver*, [27]).

Os resultados numéricos mostram que o algoritmo é capaz de fornecer soluções bem próximas das soluções “ótimas”, em um tempo razoavelmente curto, para instâncias de tamanho pequeno e médio.

³Se $G = (V, E)$ onde $V = \{1, \dots, 2k\}$ e $E = \{(1, k+1), (2, k+2), \dots, (k, 2k)\}$. Para uma ordem topológica $T = \langle 1, k+1, 2, k+2, \dots, k, 2k \rangle$, uma chamada recursiva a $SolveP(G)$ sem armazenamento de tabelas resolveria o problema $P_M(H)$ restrito ao grafo H induzido por $V(H) = \{k, 2k\}$ 3^{K-1} vezes. Por outro lado, com o armazenamento de tabelas, $P_M(H)$ é resolvido apenas uma vez.

Tabela 4.3: Qualidade de aproximação e tempo de execução para $P_M(G)$

Número de poços	M	J_M/UB (%)	Tempo de execução
6	10	99,61	0,06
	80	99,86	0,10
	320	99,90	6,36
6	20	99,15	0,04
	80	99,95	0,05
	320	99,96	0,85
12	40	99,27	0,07
	80	99,92	0,08
	320	99,99	0,73
12	40	95,91	1,80
	80	97,25	2,75
	320	98,20	4,09
24	40	92,33	3,00
	320	96,71	3,29
	640	96,99	7,67
24	40	96,10	2,12
	320	97,88	2,12
	640	97,89	3,90
36	100	95,04	2,43
	400	97,32	2,68
	800	97,57	4,96
36	100	94,77	2,62
	400	97,17	2,84
	800	97,38	5,23
Média		97,51	2,49

4.4 Extensões

Nesta seção, indicamos como seria possível resolver os problemas $P_W(F)$ e $P_W(G)$, e também $P_M(G)$ quando o grafo G contém ciclos.

Os algoritmos para a solução de $P_W(F)$ e $P_W(G)$ funcionariam seguindo os mesmos passos dos algoritmos para $P_E(F)$ e $P_E(G)$ respectivamente, bastando substituir os valores de \tilde{q}_p^n por $\text{Min}\{\tilde{q}_p^{n,k} : k = 1, \dots, \kappa(n)\}$, onde $\kappa(n)$ é o número de WPC s disponível para o poço n . Isto causaria uma penalidade em termos de tempo computacional da ordem de $K = \max\{\kappa(n) : n = 1, \dots, N\}$, ou seja, o tempo de execução de $P_W(F)$ pertenceria a $\Theta(4KNM^2)$, e o tempo de execução de $P_W(G)$ a $O(K|D|NM^2)$.

Consideremos agora o problema $P_M(G)$ quando o grafo G que representa as restrições de precedência contém ciclos. A seguir, mostramos um procedimento para resolver tal instância de $P(G)$ que invoca o algoritmo criado para resolver instâncias cujas restrições de precedência formam um

grafo acíclico. Primeiro, encontre os componentes fortemente conexos C_1, \dots, C_K de G e defina aqueles que contém mais de um poço como um “super-poço”. Como cada componente fortemente conexo C_k terá todos seus poços ativos ou inativos, seu super-poço correspondente se comportará como um único poço. Segundo, resolva uma família de problemas similares a $P(\emptyset)$ para cada super-poço, onde todos seus poços estão ativos e para cada $m \in \{0, \dots, M\}$, produzindo uma tabela com as soluções e armazenando-a no dicionário D . Terceiro, resolva um problema $P_M(G')$ onde os vértices de G' correspondem aos elementos de $\{C_1, \dots, C_K\}$ e cujo conjunto de arestas induz um grafo acíclico. Além do cálculo envolvendo os componentes conexos e o cálculo prévio de tabelas, os super-poços devem ser tratados diferentemente quando o algoritmo apresentado na Seção 4.3 for utilizado para resolver $P_M(G')$, ou seja, buscar um valor em uma tabela ao invés de resolver um problema associado a um super-poço. Do procedimento esboçado aqui, pode-se concluir que seu tempo de execução é provavelmente melhor que o tempo do algoritmo que trata de grafos acíclicos.

4.5 Sumário

Neste capítulo foram apresentados algoritmos recursivos e de Programação Dinâmica para a solução do problema de alocação ótima de gás de elevação quando os comportamentos esperados dos poços são conhecidos e existem restrições de precedência cuja representação forma uma floresta ($P_E(F)$) e um grafo acíclico ($P_E(G)$). Através de experimentos numéricos, foi constatado que o desempenho destes algoritmos é bastante satisfatório, e que são capazes de fornecer soluções quase-ótimas em tempo relativamente baixo. Os resultados desta seção foram publicados em [11], [28], [9], [29].

Capítulo 5

Solução de $P(\emptyset)$ Utilizando Programação Linear Inteira Mista

Neste capítulo, adotamos uma abordagem diferente para a resolver o problema de alocação de gás, ou seja, a linearização por partes das *WPCs* e a utilização de técnicas de solução baseadas em Programação Linear Inteira Mista. Esta mudança ocorreu pois alterar os algoritmos de Programação Dinâmica de modo a tratar novas restrições não é uma tarefa trivial. Além disso, as *WPCs* são normalmente fornecidas como um conjunto de pontos, o que torna a linearização por partes um método mais natural a ser adotado.

5.1 Linearização por Partes Aplicada a P

Inicialmente, relembramos o problema de alocação de gás de elevação sob a restrição de uma taxa máxima de injeção:

$$\begin{aligned} P : \text{Maximizar} \quad & f = \sum_{n=1}^N f_n(q_i^n, q_p^n) \\ \text{Sujeito a :} \quad & \sum_{n=1}^N q_i^n \leq q_i^{\max} \\ & q_p^n = \tilde{q}_p^n(q_i^n) \quad n = 1, \dots, N \\ & l_n y_n \leq q_i^n \leq u_n y_n \quad n = 1, \dots, N \\ & y_n \in \{0, 1\} \quad n = 1, \dots, N \end{aligned} \tag{5.1}$$

O problema P possui funções não-lineares e variáveis discretas, portanto pertence à classe de problemas de Programação Não-Linear Inteira Mista, que figuram entre os problemas de otimização

mais difíceis de serem resolvidos. Procedimentos para encontrar limites superiores para a solução ótima através da relaxação da integralidade das variáveis discretas podem dar origem a problemas difíceis de Programação Não-Linear [5], e procedimentos de enumeração das soluções podem não ser efetivos se não houverem limites apertados.

Assim, ao invés de resolver P diretamente, podemos linearizar por partes suas funções não-lineares, transformando P em um problema de Programação Linear Inteira Mista. Através desta reformulação, é possível calcular limites para as soluções através de procedimentos de Programação Inteira, ferramentas já consolidadas e de grande potencial. Esta abordagem possui desvantagens, como o aumento do número de variáveis discretas e a perda de acuracidade, porém os experimentos realizados aqui mostram que os benefícios da formulação linear inteira mista superam os aspectos negativos.

Antes de apresentar a formulação utilizando linearização por partes, algumas definições necessárias ao desenvolvimento são dadas abaixo.

Definição 5.1 $Q_i^n = \{q_i^{n,0}, q_i^{n,1}, \dots, q_i^{n,\kappa(n)}\}$ é o conjunto de taxas de injeção de gás para o poço n , onde $\kappa(n)$ é o número de taxas de injeção de gás para as quais o fluxo de saída é conhecido.

Definição 5.2 $Q_p^n = \{q_p^{n,0}, q_p^{n,1}, \dots, q_p^{n,\kappa(n)}\}$ é o conjunto de taxas de fluxo de saída para o poço n , tal que $q_p^{n,k} = q_p^n(q_i^{n,k})$ para $k = 0, \dots, \kappa(n)$.

Hipótese 5.1 Para cada poço $n \in \mathcal{N} = \{1, \dots, N\}$, $q_i^{n,0} = q_p^{n,0} = 0$, indicando que o primeiro ponto operacional é a origem; $q_i^{n,k-1} < q_i^{n,k}$ para $k = 1, \dots, \kappa(n)$, significando que os valores de taxas de injeção são distintos, $q_i^{n,1} = l_n$ e $q_i^{n,\kappa(n)} = u_n \leq q_i^{max}$, eliminando os valores de taxas de injeção desnecessários (infactíveis).

Assim, podemos seguir as diretivas fornecidas em [37] e formular o problema de alocação ótima de gás de injeção como um problema linearizado por partes:

$$\begin{aligned}
 P_{pl} : \text{Maximizar} \quad & f = \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} (f^{n,k-1} \lambda_{n,k}^L + f^{n,k} \lambda_{n,k}^R) \\
 \text{Sujeito a :} \quad & \sum_{n=1}^N q_i^n \leq q_i^{max} \\
 & \lambda_{n,k}^L, \lambda_{n,k}^R \geq 0 \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \\
 & \lambda_{n,k}^L + \lambda_{n,k}^R = x_{n,k} \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \\
 & \sum_{k=1}^{\kappa(n)} x_{n,k} = 1 \quad n = 1, \dots, N \\
 & q_i^n = \sum_{k=2}^{\kappa(n)} (q_i^{n,k-1} \lambda_{n,k}^L + q_i^{n,k} \lambda_{n,k}^R) \quad n = 1, \dots, N \\
 & x_{n,k} \in \{0, 1\}, \quad n = 1, \dots, N, \quad k = 1, \dots, \kappa(n)
 \end{aligned} \tag{5.2}$$

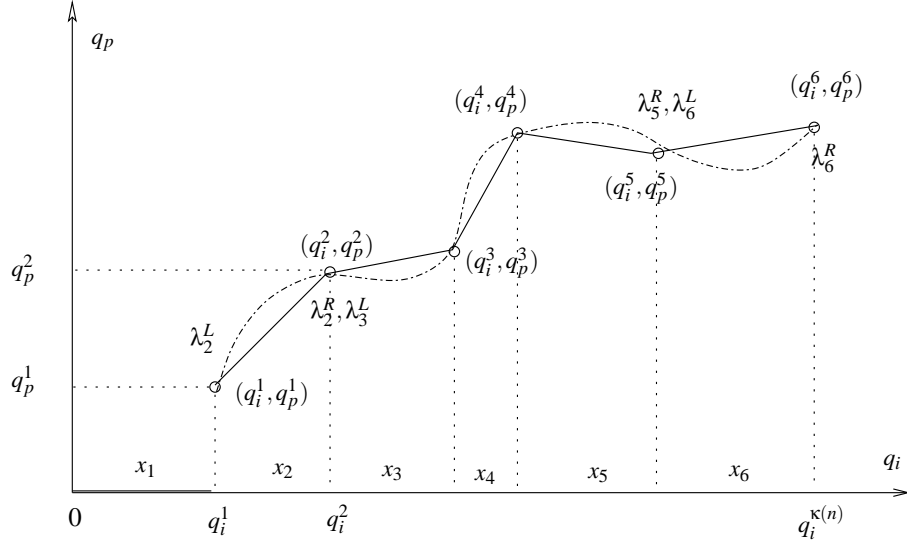


Figura 5.1: Ilustração da linearização por partes da função do fluxo de saída de um poço n utilizando o modelo de Sherali (Seção 2.3.3).

onde:

- $f^{n,k} = f_n(q_i^{n,k}, q_p^{n,k})$ é a contribuição para a função objetivo do poço n no ponto de operação $(q_i^{n,k}, q_p^{n,k})$;
- $\lambda_n = \{\lambda_{n,k}^L, \lambda_{n,k}^R : k = 2, \dots, \kappa(n)\}$ é o conjunto de fatores que induzem a combinação convexa dos elementos de Q^n ;
- $x_n = \{x_{n,k} : k = 1, \dots, \kappa(n)\}$ é o conjunto de variáveis discretas que forçam a combinação convexa de apenas dois elementos consecutivos de Q^n ; $x_{n,k}$ assume valor 1 se e somente se $q_i^n = q_i^{n,k-1} \lambda_{n,k}^L + q_i^{n,k} \lambda_{n,k}^R$ e $q_p^n = q_p^{n,k-1} \lambda_{n,k}^L + q_p^{n,k} \lambda_{n,k}^R$, com exceção de $x_{n,1}$ que assume valor 1 se e somente se $q_i^n = q_p^n = 0$, ou seja, $x_{n,k}$ assume valor 1 se $q_i^n \in [q_i^{n,k-1}, q_i^{n,k}]$ e $q_p^n \in [q_p^{n,k-1}, q_p^{n,k}]$ é aproximado linearmente neste dado intervalo pela combinação convexa de seus pontos extremos como mostra a Figura 5.1.

Para simplificar os desenvolvimentos a seguir, introduzimos a seguinte notação: $\lambda = \bigcup_{n=1}^N \lambda_n$; $x = \bigcup_{n=1}^N x_n$; $q_i = \{q_i^n : n = 1, \dots, N\}$; e $K = \sum_{n=1}^N \kappa(n)$. Utilizando esta notação, o espaço de soluções factíveis pode ser representado pelo conjunto $X_{pl} = \{(x, \lambda, q_i) \in \mathbb{B}^K \times \mathbb{R}^{2(K-N)} \times \mathbb{R}^N : (x, \lambda, q_i) \text{ satisfaz as restrições de (5.2)}\}$, onde $\mathbb{B} = \{0, 1\}$.

É sabido que na minimização (maximização) de uma função convexa (côncava) linearizada por partes, as variáveis x assumem valores inteiros naturalmente, i.e., os valores de λ diferentes de zero são sempre consecutivos [31]. Isto permite que o problema seja simplificado mais uma vez, descartando quase todas as variáveis x , sendo que apenas $x_{n,1}$ para $n = 1, \dots, N$ é necessária para determinar

se o poço está ou não ativo. Para o problema em questão, a função objetivo é separável porém não côncava, devido à região não factível entre $q_i^n = 0$ e $q_i^n = l^n$. No entanto, se a função objetivo de um poço n for côncava dentro do intervalo $[l_n, u_n]$, o número de variáveis discretas pode cair de $\kappa(n)$ para apenas uma variável.

Examinando as restrições de P_{pl} , notamos que algumas variáveis de decisão podem ser eliminadas da formulação, sendo substituídas pelas equações:

$$x_{n,1} = 1 - \sum_{k=2}^{\kappa(n)} x_{n,k} \quad n = 1, \dots, N \quad (5.3)$$

$$\lambda_{n,k}^R = x_{n,k} - \lambda_{n,k}^L \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \quad (5.4)$$

$$q_i^n = \sum_{k=2}^{\kappa(n)} q_i^{n,k} x_{n,k} + \sum_{k=2}^{\kappa(n)} (q_i^{n,k-1} - q_i^{n,k}) \lambda_{n,k}^L \quad n = 1, \dots, N \quad (5.5)$$

Substituindo as variáveis de P_{pl} pelas respectivas equações (5.3)–(5.5) obtemos uma formulação mais compacta:

$$\begin{aligned} P_{cpl} : \text{Maximizar} \quad & f = \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [f^{n,k} x_{n,k} - (f^{n,k} - f^{n,k-1}) \lambda_{n,k}^L] \\ \text{Sujeito a :} \quad & \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [q_i^{n,k} x_{n,k} - (q_i^{n,k} - q_i^{n,k-1}) \lambda_{n,k}^L] \leq q_i^{max} \\ & \lambda_{n,k}^L \geq 0 \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \\ & x_{n,k} - \lambda_{n,k}^L \geq 0 \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \\ & \sum_{k=2}^{\kappa(n)} x_{n,k} \leq 1 \quad n = 1, \dots, N \\ & x_{n,k} \in \{0, 1\} \quad n = 1, \dots, N, \quad k = 2, \dots, \kappa(n) \end{aligned} \quad (5.6)$$

Os vetores x e λ^L podem representar vetores de variáveis da mesma forma que em P_{pl} , permitindo que o espaço factível de P_{cpl} seja representado pelo conjunto $\mathcal{X}_{cpl} = \{(x, \lambda^L) \in \mathbb{B}^{K-N} \times \mathbb{R}^{K-N} : (x, \lambda^L) \text{ satisfaz as restrições de (5.6)}\}$.

5.2 Desigualdades Válidas

Um dos objetivos deste trabalho é fornecer uma formulação versátil de P e algoritmos simples e eficientes que possam ser integrados a simuladores e *software* de controle de produção de hidrocarbonetos. Neste sentido, analisamos o poliedro $\mathcal{P}_{cpl} = \text{conv}(\mathcal{X}_{cpl})$ para identificar desigualdades válidas que aceleram a busca *branch-and-bound* de algoritmos comerciais e não-comerciais. Para facilitar o entendimento do leitor, apresentamos uma instância com 6 poços na Tabela 5.1, e definimos $q_i^{max} = 6,5$. Estes dados serão utilizados para ilustrar os conceitos apresentados ao longo do texto.

Tabela 5.1: Níveis de injeção para uma instância de 6 poços, $q_i^{max} = 6, 5$.

$n \backslash k$	$q_i^{n,k}$						$\kappa(n)$
	1	2	3	4	5	6	
1	0,5	2	3	4	5	6	6
2	1	2	3	4	5	5,5	6
3	1	2	3	4	5	6	6
4	1	2	3	4	4,5		5
5	1	2	3	3,5			4
6	1	2	3	4	5		5

A seguir, introduzimos a terminologia utilizada aqui e estabelecemos condições para que \mathcal{P}_{cpl} possua dimensão cheia.

- $\mathcal{N} = \{1, \dots, N\}$ é o conjunto dos índices dos poços. Para o exemplo da Tabela 5.1, $\mathcal{N} = \{1, 2, 3, 4, 5, 6\}$;
- $\Omega = \{(n, k) : n = 1, \dots, N, k = 2, \dots, \kappa(n)\}$ é o conjunto de todos os pares (n, k) =(poço, nível de operação). No exemplo, $\Omega = \{(1, 2), \dots, (1, 6), (2, 2), \dots, (2, 6), (3, 2), \dots, (3, 6), (4, 2), \dots, (4, 5), (5, 2), \dots, (5, 4), (6, 2), \dots, (6, 5)\}$;
- $\Omega_n = \{(n, k) \in \Omega\}$ é o conjunto de todos os pares (n, k) =(poço, nível de operação) em Ω relativos ao poço n . Para $n = 5$, $\Omega_5 = \{(5, 2), (5, 3), (5, 4)\}$;
- $\Omega(U) = \bigcup_{n \in U} \Omega_n$ para $U \subseteq \mathcal{N}$ é a união dos conjuntos Ω_n para todos os poços n contidos no conjunto U . Para $U = \{3, 4\}$, $\Omega(U) = \{(3, 2), \dots, (3, 6), (4, 2), \dots, (4, 5)\}$;
- $\gamma(S) = \sum_{(n,k) \in S} q_i^{n,k-1}$ onde $S \subseteq \Omega$ é a taxa de injeção mínima necessária para ativar todos os poços n no intervalo k , sendo que o par (n, k) pertence a um conjunto S . Para $S = \{(1, 4), (2, 2), (4, 4)\}$, $\gamma(S) = q_i^{1,3} + q_i^{2,1} + q_i^{4,3} = 3 + 1 + 3 = 7$;
- $S(n, k) = S - \{(n, k)\}$ onde $S \subseteq \Omega$ são todos os pares (n, k) =(poço, nível de operação) do conjunto S exceto o par (n, k) . Para o conjunto S definido no item anterior, $S(2, 2) = \{(1, 4), (4, 4)\}$;
- $N(S) = \{n : (n, k) \in S\}$ para $S \subseteq \Omega$ é o conjunto dos índices dos poços n que possuem um par (n, k) em S . Para $S = \{(1, 4), (2, 2), (4, 4)\}$, $N(S) = \{1, 2, 4\}$; e
- $(n(S), k(S)) = \arg \max\{q_i^{n,k-1} : (n, k) \in S\}$ para $S \subseteq \Omega$ é o par (n, k) de S que necessita da maior taxa de injeção de gás para ativar o poço n no nível mínimo do intervalo k . Para $S = \{(1, 4), (2, 2), (4, 4)\}$, $(n(S), k(S)) = \arg \max\{q_i^{1,3}, q_i^{2,1}, q_i^{4,3}\} = \arg \max\{3, 1, 3\} = (1, 4)$.

Proposição 5.1 $\mathcal{P}_{cpl} = \text{conv}(\mathcal{X}_{cpl})$ possui dimensão cheia.

Prova: Seja $Z \in \mathbb{R}^{2(K-N)}$ uma matriz cujas linhas sejam vetores incidentes $z^U \in \mathcal{X}_{cpl}$, cujas entradas diferentes de zero correspondam aos elementos de $U \subseteq \{x_{n,k}, \lambda_{n,k}^L : (n,k) \in \Omega\}$. Sejam $z^{\{x_{n,k}\}}$ e $z^{\{x_{n,k}, \lambda_{n,k}^L\}}$ para $n = 1, \dots, N$ e $k = 2, \dots, \kappa(n)$ as linhas de Z , todos elementos de \mathcal{X}_{cpl} de acordo com a Hipótese 5.1. Claramente, $\text{rank}(Z) = 2(K-N)$ e, como o vetor nulo pertence a \mathcal{X}_{cpl} , concluímos que $\text{conv}(\mathcal{X}_{cpl})$ possui dimensão cheia, ou seja, $\dim(\text{conv}(\mathcal{X}_{cpl})) = 2(K-N)$. ■

Considerando que \mathcal{P}_{cpl} possui uma estrutura semelhante ao problema da mochila [42], as desigualdades de cobertura do problema da mochila podem ser reformuladas como desigualdades válidas para \mathcal{P}_{cpl} , porém neste caso os elementos da cobertura consistem de pares de poços e níveis de injeção (pares (n,k)). Os conceitos de *cobertura* e *cobertura mínima* nas variáveis x , assim como as desigualdades válidas induzidas, são dadas a seguir.

Definição 5.3 Um subconjunto $C \subseteq \Omega$ é uma cobertura¹ se $\gamma(C) > q_i^{\max}$ e $m \neq n$ para todo $(m,i), (n,j) \in C$, $(m,i) \neq (n,j)$.

Em palavras, C é uma cobertura se for um conjunto de pares $(n,k) \in \Omega$ de poços distintos tal que a taxa de injeção mínima necessária para ativar os poços em C nos níveis de operação k supera a taxa de injeção disponível. Utilizando o exemplo da Tabela 5.1, $C = \{(1,4), (2,2), (4,4)\}$ é uma cobertura, pois $|C| = |\mathcal{N}(C)| = 3$ e $\gamma(C) = 7 > 6,5 = q_i^{\max}$.

Definição 5.4 Um subconjunto $C \subseteq \Omega$ é uma cobertura mínima se:

- i) C é uma cobertura; e
- ii) $\forall (n,k) \in C$, $\gamma(Q) \leq q_i^{\max}$ onde $Q = C(n,k)$ se $k = 2$ ou então $Q = C(n,k) \cup \{(n,k-1)\}$ se $k > 2$.

C é estritamente mínima se a desigualdade do item (ii) é estrita.

A cobertura $C = \{(1,4), (2,2), (4,4)\}$ é estritamente mínima, pois:

- para $(n,k) = (1,4)$, temos $Q = C(1,4) \cup \{(1,3)\} = \{(1,3), (2,2), (4,4)\}$ com $\gamma(Q) = 6 < 6,5 = q_i^{\max}$;
- para $(n,k) = (2,2)$, temos $Q = C(2,2) = \{(1,3), (4,4)\}$ com $\gamma(Q) = 6 < 6,5 = q_i^{\max}$;
- para $(n,k) = (4,4)$, temos $Q = C(4,4) \cup \{(4,3)\} = \{(1,4), (2,2), (4,3)\}$ com $\gamma(Q) = 6 < 6,5 = q_i^{\max}$.

Seja $\Gamma_C = \{(n,j) : (n,k) \in C, j = 2, \dots, k\}$ e seja $\mathcal{P}_C = \{(x, \lambda^L) \in \mathcal{P}_{cpl} : x_{n,k} = \lambda_{n,k}^L = 0, \forall (n,k) \in \Omega - \Gamma_C\}$ a projeção de \mathcal{P}_{cpl} no espaço de Γ_C . Utilizando a cobertura C acima, $\Gamma_C = \{(1,2), (1,3), (1,4), (2,2), (4,2), (4,3), (4,4)\}$.

¹Note que qualquer cobertura C possui cardinalidade $|C| \geq 2$ de acordo com a Hipótese 5.1.

Corolário 5.1 $\dim(\mathcal{P}_{cpl}) = 2|\Gamma_C|$.

Proposição 5.2 Dada uma cobertura C , a desigualdade de cobertura

$$\sum_{(n,k) \in C} x_{n,k} \leq |C| - 1 \quad (5.7)$$

é válida para \mathcal{P}_{cpl} .

Prova: Suponha que exista (x, λ^L) em \mathcal{P}_{cpl} que viola (5.7). Então, $\sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [q_i^{n,k} x_{n,k} - (q_i^{n,k} - q_i^{n,k-1}) \lambda_{n,k}^L] \geq \sum_{(n,k) \in C} [q_i^{n,k} x_{n,k} - (q_i^{n,k} - q_i^{n,k-1}) \lambda_{n,k}^L] \geq \sum_{(n,k) \in C} q_i^{n,k-1} = \gamma(C) > q_i^{max}$ o que contradiz a primeira restrição de (5.6) e, portanto, refuta a hipótese. ■

A desigualdade de cobertura induzida por $C = \{(1,4), (2,3), (4,3)\}$ é válida para \mathcal{P}_{cpl} : $x_{1,4} + x_{2,2} + x_{4,4} \leq 2$.

Proposição 5.3 Se C é uma cobertura estritamente mínima, então $F_C = \{(x, \lambda^L) \in \mathcal{P}_C : \sum_{(n,k) \in C} x_{n,k} = |C| - 1\}$ é uma face de dimensão cheia de \mathcal{P}_C .

A maximalidade de (5.7) obtida de uma cobertura estritamente mínima C , com respeito a \mathcal{P}_C , pode ser demonstrada mostrando que uma desigualdade que induz uma face máxima F de \mathcal{P}_C , tal que $F_C \subseteq F$, difere de (5.7) por uma constante multiplicativa. A princípio, a desigualdade de cobertura (5.7) pode ser submetida a um procedimento de *lifting* exato para obter uma face de \mathcal{P}_{cpl} de dimensões maiores.

Um procedimento mais simples para fortalecer as desigualdades de cobertura consiste em adicionar pares (n, k) que requerem mais recursos do que os pares contidos em C . Isto leva a uma cobertura estendida e a desigualdade válida definida abaixo.

Definição 5.5 $E(C) = \{(n, j) : (n, k) \in C, j = k, \dots, \kappa(n)\} \cup \{(n, k) \in \Omega : q_i^{n,k-1} \geq q_i^{n(C), k(C)-1}\}$ é a cobertura estendida de uma cobertura C .

Para nosso exemplo, $E(C) = \{(1,4), (1,5), (1,6), (2,2), (2,3), (2,4), (2,5), (2,6), (4,4), (4,5)\} \cup \{(3,4), (3,5), (3,6), (5,4), (6,4), (6,5)\}$. Note que $(n(C), k(C)) = (1,4)$ com $q_i^{n(C), k(C)-1} = 3$.

Proposição 5.4 Dada uma cobertura C , a desigualdade

$$\sum_{(n,k) \in E(C)} x_{n,k} \leq |C| - 1 \quad (5.8)$$

induzida por uma cobertura estendida $E(C)$ é válida para \mathcal{P}_{cpl} .

Note que se C não for mínima, então pode existir $C' \subset C$ tal que a desigualdade (5.8) induzida por C é implicada por (5.8) induzida por C' .

Definição 5.6 Dada uma cobertura C e $h \in \{0, \dots, |C|\}$, C_h denota o subconjunto de C tal que:

- i) $|C_h| = h$; e
- ii) $\max\{q_i^{n,k-1} : (n,k) \in C - C_h\} \leq \min\{q_i^{n,k-1} : (n,k) \in C_h\}$ se $0 < h < |C|$.

Para a cobertura exemplo,

- $C_0 = \emptyset$;
- $C_1 = \{(4,4)\}$;
- $C_2 = \{(1,4), (4,4)\}$;
- $C_3 = \{(1,4), (2,2), (4,4)\}$.

Definição 5.7 Dada uma cobertura C , $(n,k) \in C$, e $h \in \{0, \dots, |C| - 1\}$, $C(n,k)_h$ denota um subconjunto de $C(n,k)$ tal que:

- i) $|C(n,k)_h| = h$; e
- ii) $\max\{q_i^{m,j-1} : (m,j) \in C(n,k) - C(n,k)_h\} \leq \min\{q_i^{m,j-1} : (m,j) \in C(n,k)_h\}$ se $0 < h < |C| - 1$.

Para a cobertura C de nosso exemplo e $(n,k) = (1,4)$,

- $C(1,4)_0 = \emptyset$;
- $C(1,4)_1 = \{(4,4)\}$;
- $C(1,4)_2 = \{(2,2), (4,4)\}$.

Uma maneira melhor de aumentar uma desigualdade de cobertura consiste em considerar as demandas de recursos para obter fatores $\pi_{n,k} \in \{0, 1, \dots, |C| - 1\}$, $(n,k) \in E(C) - C$, ao invés de usar o limite inferior trivial. Utilizando os resultados de [3], identificamos estes fatores e condições sob as quais as desigualdades resultantes definem uma face máxima de \mathcal{P}_{cpl} .

Teorema 5.1 A desigualdade

$$\sum_{(n,k) \in \Omega} \pi_{n,k} x_{n,k} \leq \pi_0 \quad (5.9)$$

onde π_0 é um inteiro positivo, é satisfeita por todo $(x, \lambda^L) \in \mathcal{P}_{cpl} = \text{conv}(\mathcal{X}_{cpl})$ se Ω pode ser particionado em $q + 1$ subconjuntos Θ_h , $h = 0, 1, \dots, q$, $1 \leq q \leq \pi_0$, tal que:

- (i) $\pi_{n,k} = h$ para todo $(n,k) \in \Theta_h$, $h = 0, 1, \dots, q$;
- (ii) $S = \bigcup_{h=1}^q \Theta_h$ é a extensão de uma cobertura mínima C , $E(C) = S$, tal que $C \subseteq \Theta_1$, e $|C| = \pi_0 + 1$;
e
- (iii) o particionamento de Ω se dá como segue:

$$\begin{aligned}\Theta_0 &= \Omega - S, \text{ onde } \Theta_0^C = \Theta_0 \cap \Omega(N(C)) \text{ e } \Theta_0^{\bar{C}} = \Theta_0 - \Theta_0^C; \\ \Theta_1 &= S - \bigcup_{h=2}^q \Theta_h, \text{ onde } \Theta_1^C = \Theta_1 \cap \Omega(N(C)) \text{ e } \Theta_1^{\bar{C}} = \Theta_1 - \Theta_1^C; \\ \Theta_h &= \Theta_h^C \cup \Theta_h^{\bar{C}}, h = 2, \dots, q; \\ \Theta_h^C &= \{(n,k) \in \Omega : (n,i) \in C, \gamma(C(n,i)_{h-1}) \leq q_i^{n,k-1} - q_i^{n,i-1} < \gamma(C(n,i)_h)\}, \\ &\quad h = 2, \dots, q; e \\ \Theta_h^{\bar{C}} &= \{(n,k) \in \Omega : n \in \mathcal{N} - N(C), \gamma(C_h) \leq q_i^{n,k-1} < \gamma(C_{h+1})\}, h = 2, \dots, q.\end{aligned}$$

Se as condições:

- (iv) para todo $(n,k) \in C$, $h = 0, 1, \dots, q$, e para todo $(n,j) \in \Theta_h^C$,

$$\gamma((C(n,k) - C(n,k)_h) \cup \{(n,j)\}) < q_i^{max}, e \quad (5.10a)$$

- (v) para cada $h = 0, 1, \dots, q$, e para todo $(n,k) \in \Theta_h^{\bar{C}}$,

$$\gamma((C - C_{h+1}) \cup \{(n,k)\}) < q_i^{max} \quad (5.10b)$$

forem satisfeitas, além de (i), (ii), e (iii), então (5.9) induz uma faceta de \mathcal{P}_{cpl} .

Prova: (Validade) Suponha que exista $\tilde{z} = (\tilde{x}, \tilde{\lambda}^L) \in \mathcal{X}_{cpl}$ que satisfaz (i), (ii), e (iii) mas viola (5.9). Para qualquer $z = (x, \lambda^L) \in \mathcal{X}_{cpl}$, seja $J_l(x) = \{(n,k) \in \Omega : x_{n,k} = l\}$, $l = 0, 1$. Defina $\hat{z} = (\hat{x}, \hat{\lambda}^L)$ como:

- (1) $\hat{x}_{n,k} = \tilde{x}_{n,k}$ e $\hat{\lambda}_{n,k}^L = \tilde{\lambda}_{n,k}^L$ caso não seja especificado por (2), (3), ou (4) abaixo.
- (2) Seja $\tilde{\Theta}_1^C = \Theta_1^C \cap J_1(\tilde{x}) - C$ e $\tilde{\Theta}_1^{\bar{C}} = \{(n,k) \in C : (n,j) \in \tilde{\Theta}_1^C\}$. Faça as seguintes mudanças em $(\hat{x}, \hat{\lambda}^L)$:

- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 0$ para todo $(n,k) \in \tilde{\Theta}_1^C$; e
- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 1$ para todo $(n,k) \in \tilde{\Theta}_1^{\bar{C}}$.

Note que $(\hat{x}, \hat{\lambda}^L) \in \mathcal{X}_{cpl}$ e $|\Theta_1 \cap J_1(\tilde{x})| = |\Theta_1 \cap J_1(\hat{x})|$.

- (3) Seja $\{\tilde{\Theta}_h^C : h = 2, \dots, q\}$ a família de subconjuntos tal que $\tilde{\Theta}_h^C = \Theta_h^C \cap J_1(\tilde{x})$, $h = 2, \dots, q$. Seja $\tilde{\Theta}^C \subseteq \{(n,k) \in C \cap J_0(\hat{x}) : n \in N(C) - N(\bigcup_{h=2}^q \tilde{\Theta}_h^C)\}$ o maior subconjunto tal que $|\tilde{\Theta}^C| \leq \sum_{h=2}^q (h - 1)|\tilde{\Theta}_h^C|$. Faça as seguintes mudanças em $(\hat{x}, \hat{\lambda}^L)$:

- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 0$ para todo $(n,k) \in \bigcup_{h=2}^q \tilde{\Theta}_h^C$;
- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 1$ para todo $(n,k) \in \{(m,j) \in C : m \in N(\bigcup_{h=2}^q \tilde{\Theta}_h^C)\}$; e
- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 1$ para todo $(n,k) \in \tilde{\Theta}^C$.

Note que $(\hat{x}, \hat{\lambda}^L)$ permanece como elemento de \mathcal{X}_{cpl} devido a (iii). Além disso, $|\tilde{\Theta}^C|$ é máxima pois, caso contrário, $\gamma(J_1(\tilde{x})) \geq \gamma(J_1(\hat{x})) \geq \gamma(C) > q_i^{max}$ o que violaria a primeira restrição de (5.6), contradizendo a hipótese.

(4) Seja $\{\tilde{\Theta}_h^C : h = 2, \dots, q\}$ a família de subconjuntos tal que:

$$\tilde{\Theta}_h^C \subseteq \Theta_h^C \cap J_1(\tilde{x}), h = 2, \dots, q, \text{ e} \quad (5.11a)$$

$$\sum_{h=2}^q h|\tilde{\Theta}_h^C| \geq \pi_0 + 1 - |J_1(\hat{x}) \cap \Theta_1|. \quad (5.11b)$$

A existência de tal família segue de

$$\begin{aligned} \sum_{(n,k) \in \Omega} \pi_{n,k} \tilde{x}_{n,k} &= \sum_{h=1}^q h|\Theta_h \cap J_1(\tilde{x})| \\ &= |\Theta_1 \cap J_1(\tilde{x})| + \sum_{h=2}^q h|\Theta_h \cap J_1(\tilde{x})| \\ &= |\Theta_1 \cap J_1(\tilde{x})| + \sum_{h=2}^q h|\tilde{\Theta}_h^C| + \sum_{h=2}^q h|\Theta_h \cap J_1(\tilde{x}) - \tilde{\Theta}_h^C| \\ &= |J_1(\hat{x}) \cap \Theta_1| + \sum_{h=2}^q h|\Theta_h \cap J_1(\tilde{x}) - \tilde{\Theta}_h^C| \\ &= |J_1(\hat{x}) \cap \Theta_1| + \sum_{h=2}^q h|\Theta_h^C \cap J_1(\tilde{x})| \end{aligned} \quad (5.12a)$$

$$> \pi_0 \text{ [pois } \tilde{z} \text{ viola (5.9) de acordo com a hipótese]} \quad (5.12b)$$

$$\geq \pi_0 + 1 \quad (5.12c)$$

De (5.12a)–(5.12c) segue que

$$\sum_{h=2}^q h|\Theta_h^C \cap J_1(\tilde{x})| \geq \pi_0 + 1 - |J_1(\hat{x}) \cap \Theta_1|. \quad (5.13)$$

Seja $\bar{\Theta}^C \subseteq \{(n,k) \in C \cap J_0(\hat{x})\}$ o maior subconjunto tal que $|\bar{\Theta}^C| \leq \sum_{h=2}^q h|\tilde{\Theta}_h^C|$. Faça as seguintes mudanças em $(\hat{x}, \hat{\lambda}^L)$:

- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 0$ para todo $(n,k) \in \bigcup_{h=2}^q \tilde{\Theta}_h^C$; e
- faça $\hat{x}_{n,k} = \hat{\lambda}_{n,k}^L = 1$ para todo $(n,k) \in \bar{\Theta}^C$.

Então segue que $\gamma(J_1(\tilde{x})) \geq \gamma(J_1(\hat{x})) \geq \gamma(C) > q_i^{max}$, violando a primeira restrição de (5.6) e contradizendo a hipótese $\tilde{z} \in \mathcal{P}_{cpl}$.

(Maximalidade) Seja $F_\pi = \{(x, \lambda^L) \in \mathcal{P}_{cpl} : \sum_{(n,k) \in \Omega} \pi_{n,k} x_{n,k} = \pi_0\}$ a face induzida por (5.9). Seja

$$\alpha^T x + \beta^T \lambda^L = \sum_{(n,k) \in \Omega} \alpha_{n,k} x_{n,k} + \sum_{(n,k) \in \Omega} \beta_{n,k} \lambda_{n,k}^L \leq \alpha_0 \quad (5.14)$$

uma desigualdade válida que induz uma face máxima $F = \{(x, \lambda^L) \in \mathcal{P}_{cpl} : \alpha^T x + \beta^T \lambda^L = \alpha_0\}$ de \mathcal{P}_{cpl} tal que $F_\pi \subseteq F$. No que segue, provamos que F_π é máxima mostrando que (5.14) e (5.9) diferem apenas de uma constante multiplicativa.

Tome qualquer $(n, k) \in \Theta_h^C, h \in \{0, \dots, q\}$, onde $(n, i) \in C$. Seja $z = z((C(n, i) - C(n, i)_h) \cup \{(n, k)\})$ e seja \hat{z} idêntico a z , exceto por $\hat{\lambda}_{n,k}^L = \lambda_{n,k}^L - \varepsilon$. Da condição (iv), z e \hat{z} pertencem a F_π para $\varepsilon > 0$ suficientemente pequeno. Como $F_\pi \subseteq F$, z e \hat{z} devem satisfazer:

$$\sum_{(l,t) \in \Omega} \alpha_{l,t} x_{l,t} + \sum_{(l,t) \in \Omega} \beta_{l,t} \lambda_{l,t}^L = \alpha_0 \quad (5.15)$$

Subtraindo (5.15) utilizando \hat{z} de (5.15) utilizando z , deduzimos que $\beta_{n,k} = 0$.

Tome qualquer $(n, k) \in \Theta_h^{\bar{C}}, h \in \{0, \dots, q\}$. Seja $z = z((C - C_{h+1}) \cup \{(n, k)\})$ e \hat{z} idêntico a z exceto por $\hat{\lambda}_{n,k}^L = \lambda_{n,k}^L - \varepsilon$. Da condição (v), z e \hat{z} pertencem a F_π para $\varepsilon > 0$ suficientemente pequeno. Para que z e \hat{z} estejam em F , z e \hat{z} devem satisfazer (5.15). Como na dedução acima, podemos concluir que $\beta_{n,k} = 0$.

Até este ponto, mostramos que $\beta_{n,k} = 0$ para todo $(n, k) \in \Omega$.

Tome qualquer $(n, i), (m, j) \in C$. Seja $z = z(C(m, j))$ e $\hat{z} = z(C(n, i))$. Como $z, \hat{z} \in F_\pi$, devem pertencer a F e satisfazer (5.15). Assim, da mesma forma que nos desenvolvimentos acima, podemos deduzir que $\pi_{n,i} = \pi_{m,j}$ para todo $(n, i), (m, j) \in C$. Então, podemos assumir sem perda de generalidade que $\pi_0 = |C| - 1$, o que implica $\pi_{n,k} = 1$ para todo $(n, k) \in C$.

Tome qualquer (n, k) de $\Theta_h^C - C, h = 0, \dots, q$, onde $(n, i) \in C$ e seja $z = z((C(n, i) - C(n, i)_h) \cup \{(n, k)\})$. Da condição (iv), segue que $z \in F_\pi$ e deve satisfazer (5.15) para que $F_\pi \subseteq F$. Consequentemente, $\alpha^T x = \pi_0 \Leftrightarrow \sum_{(m,j) \in C(n,i) - C(n,i)_h} \pi_{m,j} + \pi_{n,k} = \pi_0 \Leftrightarrow |C| - 1 - h + \pi_{n,k} = |C| - 1 \Leftrightarrow \pi_{n,k} = h$.

Tome qualquer (n, k) de $\Theta_h^{\bar{C}}, h = 0, \dots, q$, e seja $z = z((C - C_{h+1}) \cup \{(n, k)\})$. Da condição (v), $z \in F_\pi$ e deve satisfazer (5.15) para que $F_\pi \subseteq F$. Então, $\alpha^T x = \pi_0 \Leftrightarrow \sum_{(m,j) \in C - C_{h+1}} \pi_{m,j} + \pi_{n,k} = \pi_0 \Leftrightarrow |C| - (h+1) + \pi_{n,k} = |C| - 1 \Leftrightarrow \pi_{n,k} = h$.

Dos desenvolvimentos acima, concluímos que $\pi_{n,k} = h$ para cada $(n, k) \in \Theta_h, h = 0, \dots, q$, e $\pi_0 = |C| - 1$. Então, (5.9) induz uma face máxima de \mathcal{P}_{cpl} . ■

O Teorema 5.1 oferece uma maneira bastante simples de fortalecer uma cobertura mínima, pois os fatores $\pi_{n,k}$ podem ser calculados de maneira bastante trivial. Na verdade, os fatores $\pi_{n,k}$ induzem limites inferiores para os fatores de *lifting* $\beta_{n,k}$ das variáveis $(n, k) \in \Omega - \Gamma_C$. As relações dos fatores

de *lifting* $\beta_{n,k}$ com seus limites inferiores $\pi_{n,k}$ e limites superiores $\beta'_{n,k}$ serão abordadas nas seções seguintes.

Continuando com nosso exemplo, temos que:

- $q = |C| - 1 = 2$;
- $\Theta_2^{\bar{C}} = \emptyset$;
- $\Theta_1^{\bar{C}} = \{(3,4), (3,5), (3,6), (5,4), (6,4), (6,5)\}$;
- $\Theta_2^C = \{(2,5), (2,6)\}$;
- $\Theta_1^C = \{(1,4), (1,5), (1,6), (2,2), (2,3), (2,4), (4,4), (4,5)\}$;
- $\Theta_1 = \Theta_1^C \cup \Theta_1^{\bar{C}} = \{(1,4), (1,5), (1,6), (2,2), (2,3), (2,4), (3,4), (3,5), (3,6), (4,4), (4,5), (5,4), (6,4), (6,5)\}$;
- $\Theta_2 = \{(2,5), (2,6)\}$;
- $S = \bigcup_{h=1}^q \Theta_h = \bigcup_{h=1}^2 \Theta_h = E(C)$;
- $\Theta_0^C = \{(1,2), (1,3), (4,2), (4,3)\}$;
- $\Theta_0^{\bar{C}} = \{(3,2), (3,3), (5,2), (5,3), (6,2), (6,3)\}$.

Exemplificamos agora a verificação de que as condições (5.10a) e (5.10b) são válidas:

- Condição (5.10a) para $(n,k) = (2,2)$:
 - $h = 0$: $\hat{\Theta}_0^C = \emptyset$;
 - $h = 1$: $\hat{\Theta}_1^C = \{(2,2), (2,3), (2,4)\}$
 - * $\gamma((C(2,2) - C(2,2)_1) \cup \{(2,2)\}) = \gamma(\{(1,4), (2,2)\}) = 3 + 1 = 4 < q_i^{max}$
 - * $\gamma(\{(1,4), (2,3)\}) = 3 + 2 = 5 < q_i^{max}$
 - * $\gamma(\{(1,4), (2,4)\}) = 3 + 3 = 6 < q_i^{max}$;
 - $h = 2$: $\hat{\Theta}_2^C = \{(2,5), (2,6)\}$
 - * $\gamma((C(2,2) - C(2,2)_2) \cup \{(2,5)\}) = \gamma(\{(2,5)\}) = 4 < q_i^{max}$
 - * $\gamma(\{(2,6)\}) = 5 < q_i^{max}$.
- Condição (5.10b):
 - $h = 0$ e $(n,k) = (3,3)$: $\gamma((C - C_1) \cup \{(3,3)\}) = \gamma(\{(2,2), (4,4)\} \cup \{(3,3)\}) = 1 + 3 + 2 = 6 < 6,5 = q_i^{max}$;

Tabela 5.2: Limites inferiores para os fatores de *lifting*, $\pi_{n,k}$.

(n,k)	$\pi_{n,k} = h$	(n,k)	$\pi_{n,k} = h$
(1,5)	1	(1,2)	0
(1,6)	1	(1,3)	0
(2,3)	1	(4,2)	0
(2,4)	1	(4,3)	0
(2,5)	2	(3,2)	0
(2,6)	2	(3,3)	0
(4,5)	1	(5,2)	0
(3,4)	1	(5,3)	0
(3,5)	1	(6,2)	0
(3,6)	1	(6,3)	0
(5,4)	1		
(6,4)	1		
(6,5)	1		

- $h = 1$ e $(n,k) = (6,5)$: $\gamma((C - C_2) \cup \{(6,5)\}) = \gamma(\{(2,2)\} \cup \{(6,5)\}) = 1 + 4 = 5 < 6,5 = q_i^{max}$;
- para $h = 2$, $\Theta_h^{\bar{C}} = \emptyset$.

Repetindo o processo para todos os outros pares (n,k) , verifica-se que as condições (5.10a) e (5.10b) são válidas. A Tabela 5.2 contém os limites inferiores $\pi_{n,k}$ para os fatores de *lifting* obtidos através do processo acima, e a desigualdade válida correspondente é dada por:

$$x_{1,4} + x_{1,5} + x_{1,6} + x_{2,2} + x_{2,3} + x_{2,4} + x_{3,4} + x_{3,5} + x_{3,6} + x_{4,4} + x_{4,5} + x_{5,4} + x_{6,4} + x_{6,5} + 2x_{2,5} + 2x_{2,6} \leq 2.$$

Tome uma cobertura mínima C e seja $\Phi = \Omega(\mathcal{N} - N(C)) - E(C)$. Dizemos que C é uma cobertura *forte* se $\Phi = \emptyset$ ou então $\gamma(Q) \leq q_i^{max}$ para $Q = C(n(C), k(C)) \cup \{(n(\Phi), k(\Phi))\}$. Uma cobertura estritamente mínima C é *estritamente forte* se $\gamma(Q) < q_i^{max}$. Note que existiria uma cobertura C' tal que $E(C) \subset E(C')$ se C não fosse forte e, consequentemente, a desigualdade de cobertura estendida (5.8) induzida por C' seria mais forte que (5.8) induzida por C . Note também que as condições (iv) e (v) do Teorema 5.1 implicam que C é uma cobertura estritamente forte.

Condições necessárias e suficientes para que a desigualdade (5.8) induza uma faceta de \mathcal{P}_{cpl} podem ser estabelecidas. Uma das condições é que C deve ser uma cobertura estritamente forte. Apesar de induzirem facetas, estas desigualdades são menos úteis que as desigualdades (5.9) cuja eficiência será testada através de experimentos computacionais.

5.3 *Lifting* de Coberturas Mínimas

Seja C uma cobertura mínima e seja $C' = \{(n_1, k_1), \dots, (n_T, k_T)\} \subseteq \Omega - C$ um subconjunto ordenado dos pares que não aparecem em C . Seja também $\beta = \{\beta_{n,k} : (n,k) \in C'\}$ o conjunto dos fatores de *lifting* calculados resolvendo a sequência $\mathcal{K} = \{\mathcal{K}_t : t = 1, \dots, T\}$ de problemas semelhantes a P_{cpl} , recursivamente definidos por:

$$\begin{aligned} \mathcal{K}_t : \quad z_t = \text{Maximizar} \quad & \sum_{(n,k) \in C} x_{n,k} + \sum_{j=1}^{t-1} \beta_{n_j, k_j} x_{n_j, k_j} \\ \text{Sujeito a :} \quad & \sum_{(n,k) \in \Lambda_t} q_i^{n,k-1} x_{n,k} \leq q_i^{\max} \\ & \sum_{(n,k) \in \Lambda_t \cap \Omega_n} x_{n,k} \leq 1 \quad \forall n \in N(\Lambda_t) \\ & x_{n_t, k_t} = 1 \\ & x_{n,k} \in \{0, 1\} \quad \forall (n,k) \in \Lambda_t \end{aligned} \quad (5.16)$$

para $t = 1, \dots, T$ onde $\Lambda_t = \{(n_j, k_j) : j = 1, \dots, t\} \cup C$ e os coeficientes β_{n_j, k_j} são definidos como:

$$\beta_{n_j, k_j} = |C| - 1 - z_j, \quad j = 1, \dots, t-1. \quad (5.17)$$

Utilizamos agora nosso exemplo, com $C = \{(1,4), (2,2), (4,4)\}$, $C' = \{(1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (4,5), (3,4), (3,5), (3,6), (5,4), (6,4), (6,5), (1,2), (1,3), (4,2), (4,3), (3,2), (3,3), (5,2), (5,3), (6,2), (6,3)\}$, e calculamos os fatores de *lifting* para os elementos em C' , na ordem apresentada. Para ilustrar, apresentamos a seguir o problema \mathcal{K}_1 resolvido para a obtenção de $\beta_{1,5}$, onde $\Lambda_1 = \{(1,4), (2,2), (4,4), (1,5)\}$:

$$\begin{aligned} \mathcal{K}_1 : \quad z_1 = \text{Max} \quad & x_{1,4} + x_{2,2} + x_{4,4} \\ \text{Sujeito a :} \quad & 3x_{1,4} + 1x_{2,2} + 3x_{4,4} + 4x_{1,5} \leq 6,5 \\ & x_{1,4} + x_{1,5} \leq 1 \\ & x_{2,2} \leq 1 \\ & x_{4,4} \leq 1 \\ & x_{1,5} = 1 \\ & x_{n,k} \in \{0, 1\}, \forall (n,k) \in \Lambda_1 \end{aligned}$$

A solução do problema é $z_1 = 1$, e portanto $\beta_{1,5} = |C| - 1 - z_1 = 3 - 1 - 1 = 1$. Continuando o processo iterativo para os outros 12 pares (n,k) em C' , obtemos os fatores de *lifting* mostrados na Tabela 5.3. Note que os fatores $\beta_{n,k}$ encontrados são exatamente iguais aos fatores $\pi_{n,k}$ encontrados

Tabela 5.3: Fatores de *lifting* exatos, $\beta_{n,k}$.

t	(n,k)	z_t	$\beta_{n,k}$	t	(n,k)	z_t	$\beta_{n,k}$
1	(1,5)	1	1	14	(1,2)	2	0
2	(1,6)	1	1	15	(1,3)	2	0
3	(2,3)	1	1	16	(4,2)	2	0
4	(2,4)	1	1	17	(4,3)	2	0
5	(2,5)	0	2	18	(3,2)	2	0
6	(2,6)	0	2	19	(3,3)	2	0
7	(4,5)	1	1	20	(5,2)	2	0
8	(3,4)	1	1	21	(5,3)	2	0
9	(3,5)	1	1	22	(6,2)	2	0
10	(3,6)	1	1	23	(6,3)	2	0
11	(5,4)	1	1				
12	(6,4)	1	1				
13	(6,5)	1	1				

na seção anterior utilizando o Teorema 5.1 e mostrados na Tabela 5.2.

Proposição 5.5 *Se C é uma cobertura mínima, $C' \subseteq \Omega - C$ é um subconjunto ordenado, e os coeficientes em β são calculados através da solução da sequência de problemas \mathcal{K} , então a desigualdade*

$$\sum_{(n,k) \in C} x_{n,k} + \sum_{(n,k) \in C'} \beta_{n,k} x_{n,k} \leq |C| - 1 \quad (5.18)$$

é válida para \mathcal{P}_{cpl} .

Prova: (Por indução em t) Para a base, $t = 1$, existem duas possibilidades. Se $x_{n_1,k_1} = 0$, então (5.18) se torna a desigualdade de cobertura (5.7), que é válida. Se $x_{n_1,k_1} = 1$, então (5.18) é válida se $\beta_{n_1,k_1} \leq |C| - 1 - \max\{\sum_{(n,k) \in C} x_{n,k} : \text{s.a (5.16)}\} = |C| - 1 - z_1 = \beta_{n_1,k_1}$ por definição.

Para o passo de indução, $t > 1$, existem também duas possibilidades. Se $x_{n_t,k_t} = 0$, então (5.18) é válida por indução. Se $x_{n_t,k_t} = 1$, então (5.18) é válida se $\beta_{n_t,k_t} \leq |C| - 1 - \max\{\sum_{(n,k) \in C} x_{n,k} + \sum_{j=1}^{t-1} \beta_{n_j,k_j} x_{n_j,k_j} : \text{s.a (5.16)}\} = |C| - 1 - z_t = \beta_{n_t,k_t}$ por definição. ■

O procedimento descrito acima é uma extensão do procedimento de *lifting* sequencial para \mathcal{P}_{cpl} . Os coeficientes $\beta_{n,k}$ são fatores de *lifting* que dependem tanto de C como da ordem dos elementos de $C' \subseteq \Omega - C$. Apesar do cálculo dos fatores em β significar resolver problemas de complexidade comparada a \mathcal{P}_{cpl} , um limite superior $\beta'_{n,k}$ pode ser facilmente calculado para $\beta_{n,k}$.

Definição 5.8 *Dada uma cobertura C , \tilde{C}_h é um subconjunto de C tal que $h \in \{0, \dots, |C|\}$, $|\tilde{C}_h| = h$, e $\min\{q_i^{n,k-1} : (n,k) \in C - \tilde{C}_h\} \geq \max\{q_i^{n,k-1} : (n,k) \in \tilde{C}_h\}$ quando $0 < h < |C|$, i.e., \tilde{C}_h é um subconjunto dos h elementos mais “leves” de C .*

Para o nosso exemplo, com $C = \{(1,4), (2,2), (4,4)\}$, temos que $\tilde{C}_0 = \emptyset$, $\tilde{C}_1 = \{(2,2)\}$, $\tilde{C}_2 = \{(1,4), (2,2)\}$ e $\tilde{C}_3 = \{(1,4), (2,2), (4,4)\}$.

Definição 5.9 Dada uma cobertura C , $(n,k) \in C$, e $h \in \{0, \dots, |C| - 1\}$, $\tilde{C}(n,k)_h$ é um subconjunto de $C(n,k)$ tal que $|\tilde{C}(n,k)_h| = h$ e $\min\{q_i^{m,j-1} : (m,j) \in C(n,k) - \tilde{C}(n,k)_h\} \geq \max\{q_i^{m,j-1} : (m,j) \in \tilde{C}(n,k)_h\}$ quando $0 < h < |C| - 1$, i.e., $\tilde{C}(n,k)_h$ é um subconjunto dos h elementos mais “leves” de $C(n,k)$.

Para a cobertura C de nosso exemplo e $(n,k) = (1,4)$, temos que $\tilde{C}(1,4)_0 = \emptyset$, $\tilde{C}(1,4)_1 = \{(2,2)\}$ e $\tilde{C}(1,4)_2 = \{(2,2), (4,4)\}$.

Seja $\beta'_{n,k}$ o valor de $\beta_{n,k}$ se $(n,k) = (n_1, k_1)$ e seja $z'_{n,k}$ o valor de K_1 .

Proposição 5.6 Dada uma cobertura mínima C e $(n,k) \in \Omega - C$, $\beta'_{n,k} = |C| - 1 - z'_{n,k}$ onde $z'_{n,k}$ pode ser calculado como segue:

$$z'_{n,k} = \begin{cases} h^* = \arg \max_{h \in \{0, \dots, |C| - 1\}} \{\gamma(\tilde{C}_h) \leq q_i^{\max} - q_i^{n,k-1}\} & \text{se } n \in \mathcal{N} - N(C) \\ h^* = \arg \max_{h \in \{0, \dots, |C| - 1\}} \{\gamma(\tilde{C}(n,i)_h) \leq q_i^{\max} - q_i^{n,k-1}\} & \text{se } (n,i) \in C \end{cases}$$

Prova: Se $n \in N(C)$, então x' obtido quando $x'_{m,j} = 1$ para todo $(m,j) \in \tilde{C}(n,i)_{h^*} \cup \{(n,k)\}$, onde $(n,i) \in C$, e $x'_{m,j} = 0$ para todo $(m,j) \in \Omega - \tilde{C}(n,i)_{h^*} \cup \{(n,k)\}$ é factível para \mathcal{K}_1 e seu valor objetivo é $z_1 = |\tilde{C}(n,i)_{h^*}| = h^* \leq z'_{n,k}$. De fato, x' é ótimo porque não pode haver solução factível com mais de h^* elementos de C iguais a 1, portanto $z_1 = z'_{n,k}$. Similarmente, pode-se provar que isto é satisfeito se $n \in \mathcal{N} - N(C)$. ■

Em nosso exemplo, para $(n,k) = (1,5)$, $z'_{1,5} = \arg \max\{\gamma(\tilde{C}(1,4)_h) \leq 6,5 - q_i^{1,4} = 6,5 - 4 = 2,5\} \Rightarrow z'_{1,5} = 1$ e portanto $\beta'_{1,5} = |C| - 1 - z'_{n,k} = 3 - 1 - 1 = 1$. Para $(n,k) = (3,6)$, $z'_{3,6} = \arg \max\{\gamma(\tilde{C}_h) \leq 6,5 - q_i^{3,5} = 6,5 - 5 = 1,5\} \Rightarrow z'_{3,6} = 1$ e portanto $\beta'_{3,6} = 3 - 1 - 1 = 1$. Procedendo da mesma maneira para todos os pares (n,k) , obtemos os limites superiores $\beta'_{n,k}$ mostrados na Tabela 5.4. Note que, para este exemplo, os limites inferiores $\pi_{n,k}$ são iguais a $\beta'_{n,k}$ para todos os pares (n,k) , o que significa que $\pi_{n,k} = \beta'_{n,k} = \beta_{n,k}$.

Note que $\beta_{n,k}$ depende da cobertura C mas não da ordem C' . Seja C'_t a ordem obtida de C' colocando um par $(n,k) \in C'$ na t -ésima posição, e seja $\beta_{n,k}(C'_t)$ o respectivo fator de *lifting*. É evidente da estrutura do problema \mathcal{K}_t que $\beta_{n,k}(C'_t)$ é não-crescente em t e $\beta_{n,k}(C'_1) = \beta'_{n,k}$. Portanto, $\beta'_{n,k}$ é um limite superior para o fator de *lifting* de $x_{n,k}$ para qualquer sequência ordenada $C' \subseteq \Omega - C$, i.e. $\beta'_{n,k} \geq \beta_{n,k}$.

Tabela 5.4: Limites superiores para os fatores de *lifting*, $\beta'_{n,k}$.

(n, k)	$z'_{n,k}$	$\beta'_{n,k}$	(n, k)	$z'_{n,k}$	$\beta'_{n,k}$
(1,5)	1	1	(1,2)	2	0
(1,6)	1	1	(1,3)	2	0
(2,3)	1	1	(4,2)	2	0
(2,4)	1	1	(4,3)	2	0
(2,5)	0	2	(3,2)	2	0
(2,6)	0	2	(3,3)	2	0
(4,5)	1	1	(5,2)	2	0
(3,4)	1	1	(5,3)	2	0
(3,5)	1	1	(6,2)	2	0
(3,6)	1	1	(6,3)	2	0
(5,4)	1	1			
(6,4)	1	1			
(6,5)	1	1			

5.4 Sobre os Fatores de *Lifting*

Agora, mostramos que os fatores $\pi_{n,k}$ da desigualdade (5.9) são iguais ou diferem de no máximo uma unidade dos limites superiores dos fatores de *lifting* $\beta'_{n,k}$. Para estabelecer formalmente esta relação entre limites inferior e superior dos fatores de *lifting*, seja $\Theta_h^C = I_h^C \cup J_h^C$ a partição de Θ_h^C definida no Teorema 5.1, $h = 0, \dots, q$, onde $(n, k) \in \Theta_h^C$ pertence a I_h^C se:

$$\gamma(\tilde{C}(n, \delta(n))_{|C|-h-1}) + q_i^{n,k-1} \leq q_i^{max} \quad (5.19)$$

onde $(n, \delta(n)) \in C$, e caso contrário $(n, k) \in J_h^C$. Similarmente, seja $\Theta_h^{\bar{C}} = I_h^{\bar{C}} \cup J_h^{\bar{C}}$ uma partição de $\Theta_h^{\bar{C}}$ definida no Teorema 5.1, $h = 0, \dots, q$, onde $(n, k) \in \Theta_h^{\bar{C}}$ pertence a $I_h^{\bar{C}}$ se:

$$\gamma(\tilde{C}_{|C|-h-1}) + q_i^{n,k-1} \leq q_i^{max} \quad (5.20)$$

e caso contrário $(n, k) \in J_h^{\bar{C}}$. Seja $I^C = \bigcup_{h=0}^q I_h^C$, $J^C = \bigcup_{h=0}^q J_h^C$, $I^{\bar{C}} = \bigcup_{h=0}^q I_h^{\bar{C}}$ e $J^{\bar{C}} = \bigcup_{h=0}^q J_h^{\bar{C}}$. Seja também $I = I^C \cup I^{\bar{C}}$ e $J = J^C \cup J^{\bar{C}}$. Note que as condições (5.19) e (5.20) são versões não estritas das condições (iv) e (v) do Teorema 5.1.

Proposição 5.7 Para qualquer face de $\mathcal{P}_{cpl} = \text{conv}(\mathcal{X}_{cpl})$ obtida através de *lifting* sequencial a partir de uma cobertura mínima C ,

$$\beta'_{n,k} = \begin{cases} \pi_{n,k}, & (n, k) \in I \\ \pi_{n,k} + 1, & (n, k) \in J \end{cases}$$

Prova: (Caso 1) Seja $(n, k) \in I_h^C$. Se $h = 0$, de (5.19) temos $\gamma(\tilde{C}(n, \delta(n))_{|C|-h-1}) + q_i^{n, k-1} \leq q_i^{max} \Rightarrow \gamma(\tilde{C}(n, \delta(n))_{|C|-1}) \leq q_i^{max} - q_i^{n, k-1}$ onde $(n, \delta(n)) \in C$ e, portanto, da Proposição 5.6, $z'_{n, k} = |C| - 1 \Rightarrow \beta'_{n, k} = h = \pi_{n, k}$. Se $h > 0$, então da definição de $\pi_{n, k}$:

$$\gamma(C(n, \delta(n))_{h-1}) + q_i^{n, \delta(n)-1} \leq q_i^{n, k-1}$$

e como C é uma cobertura,

$$\begin{aligned} q_i^{max} < \gamma(C) &= \gamma(\tilde{C}(n, \delta(n))_{|C|-h}) + \gamma(C(n, \delta(n))_{h-1}) + q_i^{n, \delta(n)-1} \\ &\leq \gamma(\tilde{C}(n, \delta(n))_{|C|-h}) + q_i^{n, k-1} \end{aligned}$$

Desta desigualdade e porque a condição (5.19) é satisfeita, deduzimos que:

$$\gamma(\tilde{C}(n, \delta(n))_{|C|-(h+1)}) \leq q_i^{max} - q_i^{n, k-1} < \gamma(\tilde{C}(n, \delta(n))_{|C|-h})$$

e, da Proposição 5.6, concluímos que $z'_{n, k} = |C| - (h + 1) \Rightarrow \beta'_{n, k} = h = \pi_{n, k}$.

(Caso 2) Seja $(n, k) \in I_h^{\bar{C}}$. Da definição de $\pi_{n, k}$ temos que $\gamma(C_h) \leq q_i^{n, k-1}$ e como C é uma cobertura $\gamma(\tilde{C}_{|C|-h}) > q_i^{max} - q_i^{n, k-1}$. Desta desigualdade e porque a condição (5.20) é satisfeita, segue que

$$\gamma(\tilde{C}_{|C|-(h+1)}) \leq q_i^{max} - q_i^{n, k-1} < \gamma(\tilde{C}_{|C|-h})$$

e, portanto, da Proposição 5.6, $z'_{n, k} = |C| - (h + 1) \Rightarrow \beta'_{n, k} = h = \pi_{n, k}$.

(Caso 3) Seja $(n, k) \in J_h^{\bar{C}}$. Da definição de $\pi_{n, k}$ temos que $\gamma(C_{h+1}) > q_i^{n, k-1}$ e como C é uma cobertura $\gamma(\tilde{C}_{|C|-(h+2)}) \leq q_i^{max} - q_i^{n, k-1}$. Desta desigualdade e porque (5.20) não é satisfeita,

$$\gamma(\tilde{C}_{|C|-(h+2)}) \leq q_i^{max} - q_i^{n, k-1} < \gamma(\tilde{C}_{|C|-(h+1)})$$

e, da Proposição 5.6, $z'_{n, k} = |C| - (h + 2) \Rightarrow \beta'_{n, k} = h + 1 = \pi_{n, k} + 1$.

(Caso 4) Seja $(n, k) \in J_h^C$. Da definição de $\pi_{n, k}$, $\gamma(C(n, \delta(n))_h) + q_i^{n, \delta(n)-1} > q_i^{n, k-1}$ e como C é uma cobertura segue que $\gamma(\tilde{C}(n, \delta(n))_{|C|-(h+2)}) \leq q_i^{max} - q_i^{n, k-1}$. Desta desigualdade e porque a condição (5.19) não é satisfeita, deduzimos que:

$$\gamma(\tilde{C}(n, \delta(n))_{|C|-(h+2)}) \leq q_i^{max} - q_i^{n, k-1} < \gamma(\tilde{C}(n, \delta(n))_{|C|-(h+1)})$$

e, da Proposição 5.6, concluímos que $z'_{n, k} = |C| - (h + 2) \Rightarrow \beta'_{n, k} = h + 1 = \pi_{n, k} + 1$. ■

Observe que $J_0^C = \emptyset$ se C é uma cobertura mínima e $J_0^{\bar{C}} \cup J_0^C = \emptyset$ se C é forte. O significado do resultado acima é que parâmetros $\pi_{n, k}$ da desigualdade (5.9) estão distantes no máximo de 1 unidade do limite superior $\beta'_{n, k}$, qualquer que seja a desigualdade (5.18) obtida através de *lifting* sequencial da desigualdade de cobertura (5.7). Na verdade outro resultado pode ser estabelecido relacionando os

verdadeiros fatores de *lifting* $\beta_{n,k}$ e os parâmetros $\pi_{n,k}$. Para qualquer desigualdade submetida a *lifting* sequencial (5.9) obtida de uma cobertura mínima C e de acordo com qualquer ordem $C' \subseteq \Omega - C$, pode ser mostrado que $\beta_{n,k} = \pi_{n,k}$ para todo $(n,k) \in I$, e $\beta_{n,k} \in \{\pi_{n,k}, \pi_{n,k} + 1\}$ para todo $(n,k) \in J$.

Utilizamos agora nosso exemplo para ilustrar a Proposição 5.7 com dois elementos pertencentes a Θ_h^C e $\Theta_h^{\tilde{C}}$, $(1,4)$ e $(5,4)$ respectivamente:

$$\begin{aligned}\gamma(\tilde{C}(1,4)_1) + q_i^{1,3} &= 1 + 3 = 4 \leq 6, 5 = q_i^{max} \Rightarrow (1,4) \in I_1^C \\ \gamma(\tilde{C}_1) + q_i^{5,3} &= 1 + 3 = 4 \leq 6, 5 = q_i^{max} \Rightarrow (5,4) \in I_1^{\tilde{C}}\end{aligned}$$

Repetindo o raciocínio para todos os pares (n,k) restantes, concluímos que $J = \emptyset$, ou seja, $\beta'_{n,k} = \pi_{n,k}$ para todo $(n,k) \in \Omega$, corroborando os resultados mostrados nas Tabelas 5.2, 5.3 e 5.4.

5.5 Resultados Numéricos

Esta seção contém os resultados de uma série de experimentos computacionais realizados com o intuito de testar a eficiência das desigualdades de cobertura. As 16 instâncias do problema foram obtidas variando o número de poços de 32 a 128 e a taxa disponível de injeção de gás. As *WPCs* obedecem aos padrões tipicamente observados na prática. Com respeito à linearização por partes, as *WPCs* foram aproximadas por 19 segmentos, i.e., $\kappa(n) = 19$ para cada n , e satisfazem as condições impostas na Hipótese 5.1.

Para analisar o impacto das desigualdades de cobertura no desempenho de algoritmos comerciais e não-comerciais, realizamos os seguintes passos para cada instância:

- 1) resolver a instância de P_{cpl} utilizando o ILOG CPLEX Version 9.0 (ILOG Inc., 2003);
- 2) resolver iterativamente a relaxação contínua de P_{cpl} com um resolvedor de PL (programação linear), procurando repetidamente uma cobertura C e calculando fatores $\pi_{n,k}$ de modo que a desigualdade resultante (5.9) “corta” as soluções fracionárias, parando apenas quando um plano de corte não for encontrado;
- 3) resolver a instância de P_{cpl} aumentada com estes cortes utilizando ILOG CPLEX; e
- 4) repetir os passos 1 a 3 mas desta vez utilizando o *GNU MILP Solver* [21], que realiza uma busca *branch-and-bound* pura.

A Tabela 5.5 mostra os resultados obtidos dos experimentos conduzidos com o ILOG CPLEX. Para cada instância, a tabela mostra o número de poços e a taxa disponível de injeção de gás de elevação, o número de nós de *branch-and-bound*, número de iterações (operações de pivoteamento),

Tabela 5.5: Experimentos com ILOG CPLEX

N	q_i^{max}	sem cortes			com cortes		
		nós	iterações	tempo (s)	nós	iterações	tempo (s)
32	300	3.056	3.573	0,82	0	5	0,01
	500	654	866	0,35	0	6	0,01
	700	7.272	9.988	3,86	0	18	0,01
	1100	8.050	1.329	4,53	0	27	0,02
	1500	1.537	2.981	1,26	0	33	0,01
64	700	176.621	229.908	189,35	0	15	0,02
	1100	770.838	1.173.700	1.161,95	0	18	0,03
	2300	852.902	1.697.571	1.917,00	0	43	0,03
	2700	230.089	449.970	523,01	0	50	0,04
	3500	58.948	113.675	130,61	0	72	0,04
128	1100	22.724.941	28.296.248	73.949,20	0	24	0,05
	1500	21.011.913	26.031.027	54.594,99	0	28	0,02
	1900	18.514.993	23.204.919	47.346,40	0	35	0,05
	3100	43.742.914	65.654.963	115.245,77	0	65	0,06
	3500	18.769.706	31.618.407	53.069,40	0	76	0,05
	7000	815.574	2.646.970	2.976,54	0	132	0,07
Média		7.980.625	11.321.005	21.944,69	0	40	0,03

e tempo computacional para duas formulações, P_{cpl} e P_{cpl} aumentada com as desigualdades (5.9). Os experimentos foram realizados com uma estação de trabalho equipada com um microprocessador Pentium IV de 2.5GHz, 1 GB de RAM, e com sistema operacional GNU/Linux.

Os resultados indicam que as desigualdades de cobertura podem reduzir o tempo computacional e uso de memória drasticamente, demonstrando que podem acelerar a solução do problema de alocação ótima de gás de elevação.

Normalmente, os resolvidores comerciais pre-processam a instância do problema de modo a fortalecer sua formulação com cortes gerais e restrições lógicas. Sendo assim, repetimos os experimentos citados acima utilizando desta vez um algoritmo de *branch-and-bound* puro. Nossa intenção era confirmar o mérito intrínseco das desigualdades de cobertura e avaliar o potencial de uso de um *software* de domínio público. Realizamos os mesmos passos com exceção de que as instâncias de P_{cpl} , sejam puras ou aumentadas com desigualdades de cobertura, foram resolvidas com o *GNU MILP solver*.

A Tabela 5.6 mostra os resultados destes experimentos. Os experimentos corroboram os resultados obtidos com CPLEX: os cortes baseados nas coberturas diminuíram expressivamente o número de nós de *branch-and-bound*, bem como o tempo computacional. O processo de solução das instâncias marcadas com “*” e sem cortes foi interrompido após aproximadamente 100 horas de processamento.

Tabela 5.6: Experimentos com *GNU MILP solver*

N	q_i^{max}	sem cortes			com cortes		
		nós	iterações	tempo (s)	nós	iterações	tempo (s)
32	300	18.855	17.335	25,79	1	11	0,03
	500	2.661	2.632	3,52	1	18	0,03
	700	16.415	19.524	31,81	1	22	0,03
	1100	21.795	29.909	36,76	1	34	0,03
	1500	5.349	7.833	7,99	1	47	0,04
64	700	336.797	414.087	4.963,70	1	22	0,10
	1100	1.472.805	1.878.569	66.297,01	1	36	0,11
	2300	1.894.961	3.394.994	99.381,49	1	77	0,12
	2700	739.913	1.250.173	15.111,32	1	94	0,14
	3500	264.577	523.835	3.070,23	1	126	0,15
128	1100*	29.676.288	48.484.213	384.948,77	1	36	0,16
	1500*	29.565.486	49.945.930	389.406,36	1	47	0,16
	1900*	28.452.225	45.765.938	357.409,19	1	60	0,17
	3100*	28.584.317	45.797.646	357.409,52	1	99	0,18
	3500*	28.412.763	45.673.329	356.700,35	1	111	0,19
	7000	1.358.067	6.094.626	19.841,12	1	243	0,23
Média		9.426.454	15.581.285	128.415,30	1	68	0,12

5.6 Sumário

Neste capítulo, foi realizada a linearização por partes das funções não-lineares do problema de alocação ótima de taxas de injeção de gás de elevação sob uma única restrição, i.e., a taxa máxima disponível para injeção de gás. Através da análise deste novo problema foi possível encontrar famílias de desigualdades válidas que mostraram-se muito eficientes na redução do tempo de resposta de algoritmos comerciais e não-comerciais que resolvem problemas de otimização linear inteira mista. Os resultados desta seção foram publicados em [8] e [30].

Capítulo 6

Solução de $P^5(\emptyset)$ Utilizando Programação Linear Inteira Mista

Neste capítulo, são inseridas novas restrições ao problema tratado no Capítulo 5, de modo a torná-lo mais abrangente. Mais especificamente, abordamos a versão do problema de alocação ótima de gás de elevação sob múltiplas restrições nas instalações de superfície, chamado $P^5(\emptyset)$. Novamente, as curvas de desempenho de *gas-lift* são linearizadas por partes (o Modelo I mostrado no Capítulo 2 é utilizado pois o Modelo III ainda não era de nosso conhecimento) e desigualdades válidas são obtidas e introduzidas na formulação. A eficiência destas desigualdades na aceleração do processo de solução é testada e comprovada através de experimentos numéricos.

6.1 Formulação do Problema

O problema $P^5(\emptyset)$ ¹ de alocação ótima de uma taxa disponível de injeção de gás de elevação, sujeito a restrições nas capacidades de separação, armazenamento e tratamento de fluidos é formulado

¹O superscrito 5 denota o número de restrições representadas pela formulação.

como:

$$P^5(\emptyset) : \text{Maximizar } f = \sum_{n=1}^N (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) q_p^n - p_i q_i^n \quad (6.1a)$$

Sujeito a:

$$q_p^n = \tilde{q}_p^n(q_i^n), \quad n = 1, \dots, N \quad (6.1b)$$

$$\sum_{n=1}^N q_i^n \leq q_i^{\max} \quad (6.1c)$$

$$\sum_{n=1}^N q_p^n \leq q_p^{\max} \quad (6.1d)$$

$$\sum_{n=1}^N q_o^n = \sum_{n=1}^N \gamma_o^n q_p^n \leq q_o^{\max} \quad (6.1e)$$

$$\sum_{n=1}^N q_g^n = \sum_{n=1}^N \gamma_g^n q_p^n \leq q_g^{\max} \quad (6.1f)$$

$$\sum_{n=1}^N q_w^n = \sum_{n=1}^N \gamma_w^n q_p^n \leq q_w^{\max} \quad (6.1g)$$

$$l_n y_n \leq q_i^n \leq u_n y_n, \quad n = 1, \dots, N \quad (6.1h)$$

$$y_n \in \{0, 1\}, \quad n = 1, \dots, N \quad (6.1i)$$

onde:

- f é a função objetivo a ser maximizada;
- γ_o^n , γ_g^n e γ_w^n são as frações produzidas de óleo, gás e água respectivamente, sendo que $\gamma_o^n + \gamma_g^n + \gamma_w^n = 1$;
- p_o e p_g representam respectivamente o lucro obtido com a venda de cada unidade de volume de óleo e gás, p_w é o custo de tratamento por unidade de volume de água produzida, e p_i representa o custo da injeção de gás por unidade de volume por tempo;
- N é o número de poços que operam via *gas-lift*;
- q_p^n é a taxa de produção de fluidos de um poço n como função de q_i^n , onde $\tilde{q}_p^n(q_i^n)$ é a *WPC* do poço n ;
- q_i^n é a taxa de injeção de gás alocada para o poço n ;
- q_i^{\max} , q_p^{\max} , q_o^{\max} , q_g^{\max} e q_w^{\max} são respectivamente a taxa de injeção de gás disponível, a taxa de produção total máxima (representando a capacidade de separação de fluidos), e as taxas máximas de produção de óleo, gás e água (representando os limites de armazenamento e/ou tratamento destes fluidos);

- y_n é uma variável binária que assume valor 1 se o poço n está ativo, e 0 caso contrário;
- l_n e u_n são respectivamente os limites inferior e superior para a taxa de injeção do poço n , impostos por questões técnicas ou estratégicas.

Dado um conjunto $Q^n = \{(q_i^{n,0}, q_p^{n,0}), \dots, (q_i^{n,\kappa(n)}, q_p^{n,\kappa(n)})\}$ de $\kappa(n) + 1$ pontos de WPC para cada poço $n = 1, \dots, N$, com $(q_i^{n,0}, q_p^{n,0}) = (0, 0)$ e $q_i^{n,k-1} < q_i^{n,k}$ para $k = 1, \dots, \kappa(n)$, a WPC $\tilde{q}_p^n(q_i^n)$ pode ser representada como uma função linear por partes utilizando o Modelo I (Seção 2.3.1). Introduzindo variáveis binárias x e variáveis contínuas λ , é possível representar qualquer ponto da função linear por partes como:

$$x_{n,k} = \begin{cases} 1 & \text{se } q_i^{n,k-1} \leq q_i^n \leq q_i^{n,k} \\ 0 & \text{caso contrário} \end{cases} \quad (6.2a)$$

$$q_i^n = \lambda_{n,k-1} q_i^{n,k-1} + \lambda_{n,k} q_i^{n,k} \quad (6.2b)$$

$$q_p^n = \lambda_{n,k-1} q_p^{n,k-1} + \lambda_{n,k} q_p^{n,k} \quad (6.2c)$$

$$\lambda_{n,k-1} + \lambda_{n,k} = 1 \quad (6.2d)$$

$$\lambda_{n,k-1}, \lambda_{n,k} \geq 0 \quad (6.2e)$$

Linearizando as WPC s como mostrado nas equações (6.2a)–(6.2e) e ilustrado pela Figura 6.1, o problema $P^5(\emptyset)$ ou simplesmente P^5 pode ser reformulado como um problema de programação linear

inteira mista:

$$P_{pl}^5 : \text{Maximizar } f = \sum_{n=1}^N f_n(q_i^n, q_p^n) \quad (6.3a)$$

Sujeito a:

$$\sum_{n=1}^N q_y^n \leq q_y^{max}, \quad y = i, p, o, g, w \quad (6.3b)$$

Para $n = 1, \dots, N$:

$$\lambda_{n,k} \geq 0, \quad k = 0, \dots, \kappa(n) \quad (6.3c)$$

$$\lambda_{n,0} \leq x_{n,1} \quad (6.3d)$$

$$\lambda_{n,1} \leq x_{n,2} \quad (6.3e)$$

$$\lambda_{n,k} \leq x_{n,k} + x_{n,k+1}, \quad k = 2, \dots, \kappa(n) - 1 \quad (6.3f)$$

$$\lambda_{n,\kappa(n)} \leq x_{n,\kappa(n)} \quad (6.3g)$$

$$\sum_{k=0}^{\kappa(n)} \lambda_{n,k} = 1 \quad (6.3h)$$

$$\sum_{k=1}^{\kappa(n)} x_{n,k} = 1 \quad (6.3i)$$

$$q_i^n = \sum_{k=0}^{\kappa(n)} q_i^{n,k} \lambda_{n,k} \quad (6.3j)$$

$$q_p^n = \sum_{k=0}^{\kappa(n)} q_p^{n,k} \lambda_{n,k} \quad (6.3k)$$

$$q_y^n = \gamma_y^n q_p^n, \quad y = o, g, w \quad (6.3l)$$

$$q_i^n, q_p^n, q_o^n, q_g^n, q_w^n \in \mathbb{R} \quad (6.3m)$$

$$\lambda_{n,k} \in \mathbb{R}, \quad k = 0, \dots, \kappa(n) \quad (6.3n)$$

$$x_{n,k} \in \{0, 1\}, \quad k = 1, \dots, \kappa(n) \quad (6.3o)$$

onde:

- a função (6.3a) é a função objetivo a ser maximizada;
- a desigualdade (6.3b) representa as restrições nas instalações de superfície, i.e., os limites impostos pela taxa de injeção disponível (q_i^{max}), capacidade de separação (q_p^{max}), e armazenamento/tratamento de óleo, gás e água (q_o^{max} , q_g^{max} e q_w^{max} respectivamente);
- a restrição (6.3c) faz com que as variáveis $\lambda_{n,k}$ sejam não-negativas, enquanto (6.3d)-(6.3g) forçam as variáveis $\lambda_{n,k}$ a serem consecutivas, e (6.3h) significa que sua soma deve ser igual a um;

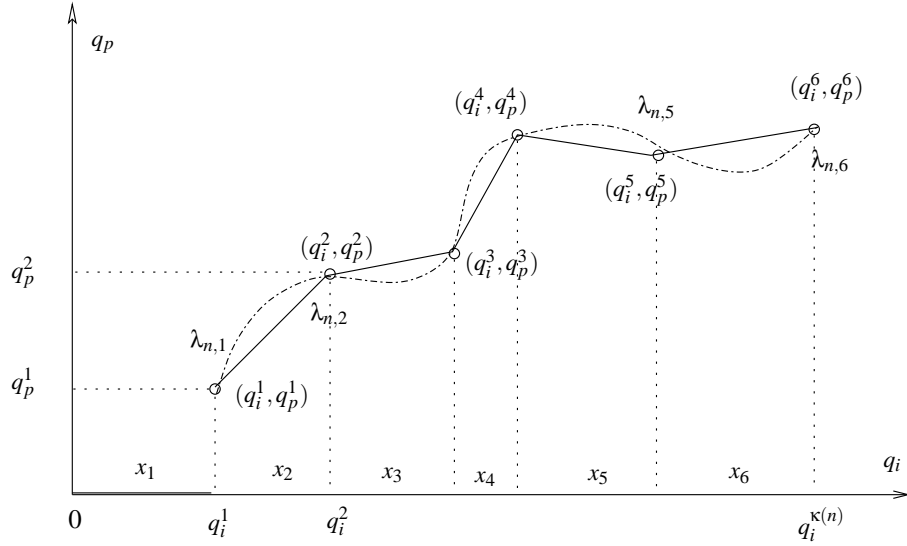


Figura 6.1: Ilustração da linearização por partes da função do fluxo de saída de um poço n utilizando o modelo de pesos por ponto (Seção 2.3.1).

- a equação (6.3i) significa que a taxa de injeção de gás para um poço n deve pertencer apenas a um determinado intervalo k ;
- as equações (6.3j) e (6.3k) definem a taxa de injeção de gás e a taxa de produção de um poço n como a combinação convexa dos pontos discretos $q_i^{n,k}$ e $q_p^{n,k}$ respectivamente; e
- a equação (6.3l) expressa as taxas de óleo, gás e água produzidas pelo poço n como funções da taxa de produção e de suas respectivas frações.

Para cada poço $n \in \mathcal{N} = \{1, \dots, N\}$, os limites l_n e u_n dados pela desigualdade (6.1h) podem ser modelados definindo $q_i^{n,1} = l_n$ e $q_i^{n,\kappa(n)} = u_n$, eliminando assim pontos que estejam fora da zona operacional. Não existe taxa de injeção factível entre $q_i^{n,0} = 0$ e $q_i^{n,1}$, ou seja, $x_{n,1} = 1$ implica que $\lambda_{n,0} = 1$ e $q_i^n = 0$, e portanto o poço n não recebe injeção de gás. Assumimos também que $q_i^{n,\kappa(n)} \leq q_i^{max}$ e que $\max\{q_p^{n,k} : k = 1, \dots, \kappa(n)\} \leq \min\{q_p^{max}, \frac{q_o^{max}}{\gamma_o^n}, \frac{q_g^{max}}{\gamma_g^n}, \frac{q_w^{max}}{\gamma_w^n}\}$ para todo $n \in \mathcal{N}$, eliminando pontos desnecessários.

Como citado no Capítulo 5, na maximização de uma função côncava linear por partes não é necessário forçar que as variáveis x sejam inteiras, i.e., as variáveis λ diferentes de zero são sempre consecutivas [31]. Sendo assim, o problema pode ser simplificado eliminando as variáveis binárias x e resolvido de forma eficiente com algoritmos de programação linear. Para o problema P_{pl}^5 , a função objetivo é separável mas não-côncava devido à região inactível entre $q_i^{n,0}$ e $q_i^{n,1}$. Além disso, a WPC linearizada por partes não é necessariamente côncava. Contudo, se a função objetivo de um poço n é côncava no intervalo $[q_i^{n,1}, q_i^{n,\kappa(n)}]$, o número de variáveis discretas cai de $\kappa(n)$ para 1. Em outras palavras, $x_{n,1}$ seria a única variável necessária, assumindo valor 1 se o poço n estivesse inativo, e caso

contrário assumindo valor 0. As condições para que esta simplificação possa ser realizada são dadas em [10], [7].

Uma formulação mais compacta de P_{pl}^5 pode ser obtida eliminando as variáveis redundantes e igualdades implícitas como segue:

$$x_{n,1} = 1 - \sum_{k=2}^{\kappa(n)} x_{n,k} \quad (6.4a)$$

$$\lambda_{n,0} = x_{n,1} \quad (6.4b)$$

$$\lambda_{n,1} = 1 - \lambda_{n,0} - \sum_{k=2}^{\kappa(n)} \lambda_{n,k} = \sum_{k=2}^{\kappa(n)} (x_{n,k} - \lambda_{n,k}) \quad (6.4c)$$

$$q_i^n = \sum_{k=2}^{\kappa(n)} \left(q_i^{n,1} x_{n,k} + (q_i^{n,k} - q_i^{n,1}) \lambda_{n,k} \right) \quad (6.4d)$$

$$q_p^n = \sum_{k=2}^{\kappa(n)} \left(q_p^{n,1} x_{n,k} + (q_p^{n,k} - q_p^{n,1}) \lambda_{n,k} \right) \quad (6.4e)$$

$$q_y^n = \gamma_y^n \sum_{k=2}^{\kappa(n)} \left(q_p^{n,1} x_{n,k} + (q_p^{n,k} - q_p^{n,1}) \lambda_{n,k} \right), \quad y = o, g, w \quad (6.4f)$$

Substituindo as expressões (6.4a)–(6.4f) por suas variáveis correspondentes em P_{pl}^5 , a seguinte formulação compacta é obtida:

$$P_{cpl}^5 : \text{Maximizar } f = \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} (f_{n,1}x_{n,k} + (f_{n,k} - f_{n,1})\lambda_{n,k}) \quad (6.5a)$$

Sujeito a :

$$\text{Para } y = i, p, o, g, w : \quad (6.5b)$$

$$\sum_{n=1}^N \sum_{k=2}^{\kappa(n)} (q_y^{n,1}x_{n,k} + (q_y^{n,k} - q_y^{n,1})\lambda_{n,k}) \leq q_y^{max}$$

Para $n = 1, \dots, N$:

$$\sum_{k=2}^{\kappa(n)} x_{n,k} \leq 1 \quad (6.5c)$$

$$\sum_{k=2}^{\kappa(n)} x_{n,k} - \sum_{k=2}^{\kappa(n)} \lambda_{n,k} \geq 0 \quad (6.5d)$$

$$\sum_{k=3}^{\kappa(n)} x_{n,k} - \sum_{k=2}^{\kappa(n)} \lambda_{n,k} \leq 0 \quad (6.5e)$$

Para $k = 2, \dots, \kappa(n) - 1$:

$$\lambda_{n,k} \leq x_{n,k} + x_{n,k+1} \quad (6.5f)$$

$$\lambda_{n,\kappa(n)} \leq x_{n,\kappa(n)} \quad (6.5g)$$

Para $k = 2, \dots, \kappa(n)$:

$$\lambda_{n,k} \geq 0 \quad (6.5h)$$

$$\lambda_{n,k} \in \mathbb{R} \quad (6.5i)$$

$$x_{n,k} \in \{0, 1\} \quad (6.5j)$$

onde $f_{n,k} = f_n(q_i^{n,k}, q_p^{n,k})$, $q_o^{n,k} = \gamma_o^n q_p^{n,k}$, $q_g^{n,k} = \gamma_g^n q_p^{n,k}$ e $q_w^{n,k} = \gamma_w^n q_p^{n,k}$, implicando em $q_o^{n,k} + q_g^{n,k} + q_w^{n,k} = (\gamma_o^n + \gamma_g^n + \gamma_w^n)q_p^{n,k} = q_p^{n,k}$. Note que a desigualdade (6.5c), a restrição (6.5j) e a igualdade (6.4a) garantem que $x_{n,1} \in \{0, 1\}$ e, portanto, $\lambda_{n,0} \in \{0, 1\}$ como definido na equação (6.4b). Além disso, a desigualdade (6.5d) garante que $\lambda_{n,1} \geq 0$, enquanto a desigualdade (6.5e) e a restrição (6.5j) garantem que $\lambda_{n,1} \leq 1$.

Os problemas de otimização P^5 , P_{pl}^5 e P_{cpl}^5 pertencem à classe de problemas NP-Difíceis² [16]. Portanto, não há e as evidências da literatura até o momento indicam que não haverá algoritmo capaz de resolver P^5 em tempo polinomial. Tempo computacional não é considerado um problema para a indústria de petróleo atual, devido à dinâmica lenta do mercado e aos longos períodos entre testes dos poços. No entanto, a tecnologia de medidores de pressão de fundo e sensores multifásicos está se tornando mais avançada, indicando que o controle automático dos poços é uma tendência promissora.

²Isto pode ser mostrado via redução de um problema da mochila para P^5 , como nos capítulos anteriores.

Sendo assim, algoritmos rápidos se tornam imperativos caso a otimização em tempo real se torne necessária.

6.2 Desigualdades Válidas para $P^5(\emptyset)$

Para facilitar a leitura, esta seção apresenta um número limitado de resultados teóricos e não apresenta todas as demonstrações, visto que muitas delas seguem os mesmos moldes daquelas contidas no Capítulo 5.

Para simplificar os desenvolvimentos a seguir, introduzimos a seguinte notação:

- $x_n = \{x_{n,k} : k = 2, \dots, \kappa(n)\};$
- $x = \{x_n : n = 1, \dots, N\};$
- $\lambda_n = \{\lambda_{n,k} : k = 2, \dots, \kappa(n)\};$
- $\lambda = \{\lambda_n : n = 1, \dots, N\};$ e
- $K = \sum_{n=1}^N \kappa(n).$

Utilizando esta notação, o espaço de soluções factíveis pode ser expresso pelo poliedro $\mathcal{P}_{cpl}^5 = \text{conv}\{(x, \lambda) \in \mathbb{B}^{K-N} \times \mathbb{R}^{K-N} : (x, \lambda) \text{ satisfaz as restrições (6.5b)–(6.5h)}\}$, onde $\mathbb{B} = \{0, 1\}$. Dado um conjunto de vetores X , $\text{conv}\{X\}$ denota o conjunto de vetores que podem ser obtidos por combinação convexa de um subconjunto finito de X , ou seja, $\text{conv}\{X\} = \{x : x = \sum_{j=1}^m \lambda_j x_j, x_j \in X \text{ e } \lambda_j \geq 0 \text{ para } j = 1, \dots, m, \text{ e } \sum_{j=1}^m \lambda_j = 1\}$. Pode-se mostrar que o poliedro \mathcal{P}_{cpl}^5 que representa a formulação P_{cpl}^5 possui dimensão cheia ($\dim(\mathcal{P}_{cpl}^5) = 2(K - N)$), i.e., não existem dependências lineares entre as variáveis.

Para ilustrar os conceitos apresentados nesta seção, utilizaremos uma instância de 3 poços mostrada nas Tabelas 6.1 a 6.3. Consideramos também que os seguintes valores limite: $q_i^{max} = 10$, $q_p^{max} = 10$, $q_o^{max} = 7$, $q_g^{max} = 1, 2$ e $q_w^{max} = 1$.

A semelhança de P_{pl}^5 com o problema da mochila [22] permite estender as *desigualdades de cobertura* descobertas em [4] e [3], e obter desigualdades válidas para \mathcal{P}_{cpl}^5 , adicionando-as à formulação e acelerando o processo de solução. Os resultados a seguir são válidos para $y = i, p, o, g$ e w onde $q_o^{n,k} = \gamma_o^n q_p^{n,k}$, $q_g^{n,k} = \gamma_g^n q_p^{n,k}$ e $q_w^{n,k} = \gamma_w^n q_p^{n,k}$. Além disso, sejam:

- $\Omega = \{(n, k) : n = 1, \dots, N, k = 2, \dots, \kappa(n)\}$ é o conjunto de todos os pares (poço, ponto de operação). Em nosso exemplo, $\Omega = \{(1, 1), \dots, (1, 5), (2, 1), \dots, (2, 5), (3, 1), \dots, (3, 5)\};$

Tabela 6.1: Poço 1

y	γ_y^1	$q_y^{1,1}$	$q_y^{1,2}$	$q_y^{1,3}$	$q_y^{1,4}$	$q_y^{1,5}$
i		1	2	5	7	10
p		3	3,3	3,4	4	2
o	0,7	2,1	2,31	2,38	2,8	1,4
g	0,2	0,6	0,66	0,68	0,8	0,4
w	0,1	0,3	0,33	0,34	0,4	0,2

Tabela 6.2: Poço 2

y	γ_y^2	$q_y^{2,1}$	$q_y^{2,2}$	$q_y^{2,3}$	$q_y^{2,4}$	$q_y^{2,5}$
i		1	2,8	4	6	10
p		1	2	5	7	10
o	0,8	0,8	1,6	4	5,6	8
g	0,1	0,1	0,2	0,5	0,7	1
w	0,1	0,1	0,2	0,5	0,7	1

Tabela 6.3: Poço 3

y	γ_y^3	$q_y^{3,1}$	$q_y^{3,2}$	$q_y^{3,3}$	$q_y^{3,4}$	$q_y^{3,5}$
i		2	4	8	9	10
p		1	3	10	4	6
o	0,8	0,8	2,4	8	3,2	4,8
g	0,1	0,1	0,3	1	0,4	0,6
w	0,1	0,1	0,3	1	0,4	0,6

- $\gamma_y(S) = \sum_{(n,k) \in S} \min(q_y^{n,k-1}, q_y^{n,k})$ para $S \subseteq \Omega$ e $y \in \{i, p, o, g, w\}$ é a mínima quantidade de recurso y necessária para ativar todos os pares (poço, ponto de operação) em S . Considerando $S = \{(1,5), (2,4)\}$ e $y = i$, temos $\gamma_i(S) = \min(q_i^{1,4}, q_i^{1,5}) + \min(q_i^{2,3}, q_i^{2,4}) = 7 + 4 = 11$; e
- $S(n,k) = S - \{(n,k)\}$ para qualquer $S \subseteq \Omega$. Para $S = \{(1,5), (2,4)\}$, $S(1,5) = \{(2,4)\}$.

Definição 6.1 Um subconjunto $C_y \subseteq \Omega$ é uma cobertura- y se $\gamma_y(C_y) > q_y^{max}$ e $m \neq n$ para todo $(m,i), (n,j) \in C_y$.

Em palavras, os pares (n,k) de uma cobertura- y pertencem a poços distintos e não podem ser ativados simultaneamente pois excederiam a capacidade de recursos disponível. Uma cobertura- i para nosso exemplo seria $C_i = \{(1,5), (2,4)\}$, pois $\gamma_i(C_i) = 11 > 10 = q_i^{max}$.

Definição 6.2 Uma cobertura- y C_y é mínima se, para todo $(n,k) \in C_y$, $\gamma_y(C_y(n,k)) \leq q_y^{max}$ e $\gamma_y(C_y(n,k) \cup \{(n,j)\}) \leq q_y^{max}$ para cada $j \neq k$ tal que $\min(q_y^{n,j-1}, q_y^{n,j}) < \min(q_y^{n,k-1}, q_y^{n,k})$.

Continuando com o exemplo, $C_i = \{(1,5), (2,4)\}$ é mínima pois:

- para $(n,k) = (1,5)$: $\gamma_i(C_i(1,5)) = \min(q_i^{2,3}, q_i^{2,4}) = 4 \leq 10 = q_i^{max}$ e $\gamma_i(C_i(1,5) \cup \{(1,j)\}) \leq 10 = q_i^{max}$ para $j \in \{2,3,4\}$;
- para $(n,k) = (2,4)$: $\gamma_i(C_i(2,4)) = \min(q_i^{1,4}, q_i^{1,5}) = 7 \leq 10 = q_i^{max}$ e $\gamma_i(C_i(2,4) \cup \{(2,j)\}) \leq 10 = q_i^{max}$ para $j \in \{2,3\}$;

Seja $q_y(S)_{max} = \max\{\min(q_y^{n,k-1}, q_y^{n,k}) : (n,k) \in S\}$ a maior quantidade do recurso mínimo y requerido por um elemento de $S \subseteq \Omega$.

Definição 6.3 Dada uma cobertura- y C_y , sua cobertura estendida é $E(C_y) = E_C(C_y) \cup E_{\bar{C}}(C_y)$ onde $E_C(C_y) = \{(n,j) \in \Omega : (n,k) \in C_y, \min(q_y^{n,j}, q_y^{n,j-1}) \geq \min(q_y^{n,k}, q_y^{n,k-1})\}$ e $E_{\bar{C}}(C_y) = \{(n,k) \in \Omega : \min(q_y^{n,k-1}, q_y^{n,k}) \geq q_y(C_y)_{max}\}$.

Utilizando os dados do exemplo, a cobertura estendida $E(C_i)$ fica dividida em $E_C(C_i) = \{(1,5), (2,4), (2,5)\}$ e $E_{\bar{C}}(C_i) = \{(3,4), (3,5)\}$. Note que $C_y \subseteq E_C(C_y)$.

Proposição 6.1 Dada uma cobertura- y C_y , a desigualdade:

$$\sum_{(n,k) \in E(C_y)} x_{n,k} \leq |C_y| - 1 \quad (6.6)$$

é válida para P_{cpl}^5 .

A desigualdade de cobertura baseada na cobertura estendida $E(C_i)$ de nosso exemplo é:

$$x_{1,5} + x_{2,4} + x_{2,5} + x_{3,4} + x_{3,5} \leq 1. \quad (6.7)$$

Da mesma maneira que as coberturas- y para as variáveis x , coberturas- y e desigualdades válidas podem ser obtidas para as variáveis λ .

Definição 6.4 Dada uma cobertura- y C_y , a cobertura- y estendida para as variáveis λ é definida por:

$$\begin{aligned} E_\lambda(C_y) = & \{(n, k) \in E(C_y) : k < \kappa(n), (n, k+1) \in E(C_y)\} \cup \\ & \{(n, 1) : (n, 2) \in E(C_y)\} \cup \\ & \{(n, \kappa(n)) \in E(C_y)\} \end{aligned}$$

Proposição 6.2 Dada uma cobertura- y C_y , a desigualdade:

$$\sum_{(n,k) \in E_\lambda(C_y)} \lambda_{n,k} \leq |C_y| - 1 \quad (6.8)$$

é válida para P_{cpl}^5 sendo que $\lambda_{n,1}$ é substituída pela Equação (6.4c).

Desigualdades válidas podem ser *fortalecidas* se sujeitas ao procedimento de *lifting* [31], [42], i.e., desigualdades podem ser estendidas para dimensões maiores incorporando as variáveis restantes. Considere uma cobertura- y C_y e sua desigualdade válida $\sum_{(n,k) \in C_y} x_{n,k} \leq |C_y| - 1$. Note que as variáveis $x_{n,k}$ com $(n, k) \in \Omega - C_y$ não aparecem na desigualdade ou, em outras palavras, seus coeficientes são zero. O processo de *lifting* da desigualdade consiste em calcular o máximo $\beta_{n,k}(C_y) \geq 0$ para cada variável $x_{n,k}$, $(n, k) \in \Omega - C_y$, tal que a desigualdade $\sum_{(n,k) \in C_y} x_{n,k} + \sum_{(n,k) \in \Omega - C_y} \beta_{n,k}(C_y) x_{n,k} \leq |C_y| - 1$ permanece válida para P_{cpl}^5 . A desigualdade de cobertura- y estendida (6.6) é uma maneira aproximada para fortalecer desigualdades válidas³. Os fatores de *lifting* $\beta_{n,k}(C_y)$ dependem da cobertura C_y e da ordem dos elementos de $\Omega - C_y$.

O cálculo de fatores de *lifting* exatos $\beta_{n,k}(C_y)$ é um problema difícil com complexidade comparável à solução do próprio P_{cpl}^5 . Sendo assim, ao invés de calcular fatores exatos, propomos um procedimento heurístico para encontrar limites inferiores $\alpha_{n,k}(C_y)$ para os fatores exatos, $0 \leq \alpha_{n,k}(C_y) \leq \beta_{n,k}(C_y)$. Um método prático para acelerar a busca por uma solução ótima para P_{cpl}^5 pode ser obtido utilizando os limites inferiores $\alpha_{n,k}(C_y)$ para fortalecer as desigualdades válidas de cobertura.

Definição 6.5 Seja C_y uma cobertura- y .

³O fator multiplicativo 1 é usado no lugar de $\beta_{n,k}(C_y)$ para todo $x_{n,k}$, $(n, k) \in E(C_y) - C_y$, enquanto as variáveis restantes são multiplicadas por zero.

- Para $h \in \{0, \dots, |C_y|\}$, $C_{y,h}$ é um subconjunto de C_y tal que $|C_{y,h}| = h$ e $\max\{\min(q_y^{n,k-1}, q_y^{n,k}) : (n,k) \in C_y - C_{y,h}\} \leq \min\{\min(q_y^{n,k-1}, q_y^{n,k}) : (n,k) \in C_{y,h}\}$ se $0 < h < |C_y|$, i.e., $C_{y,h}$ é o subconjunto com os h elementos mais “pesados” de C_y .
- Para $(n,k) \in C_y$ e $h \in \{0, \dots, |C_y| - 1\}$, $C_{y,h}(n,k)$ é um subconjunto de $C_y(n,k)$ tal que $|C_{y,h}(n,k)| = h$ e $\max\{\min(q_y^{m,j-1}, q_y^{m,j}) : (m,j) \in C_y(n,k) - C_{y,h}(n,k)\} \leq \min\{\min(q_y^{m,j-1}, q_y^{m,j}) : (m,j) \in C_{y,h}(n,k)\}$ se $0 < h < |C_y| - 1$, i.e., $C_{y,h}(n,k)$ é o subconjunto dos h elementos mais “pesados” de $C_y(n,k)$.

Em nosso exemplo:

- $C_{i,0} = \emptyset$; $C_{i,1} = \{(1,5)\}$; $C_{i,2} = \{(1,5), (2,4)\}$;
- $C_{i,0}(1,5) = C_{i,0}(2,4) = \emptyset$; $C_{i,1}(1,5) = \{(2,4)\}$; $C_{i,1}(2,4) = \{(1,5)\}$.

Definição 6.6 Dada uma cobertura C_y , o limite inferior $\alpha_{n,k}(C_y)$ para o fator de lifting de cada $(n,k) \in E(C_y) - C_y$ é obtido como segue:

- se $n \in \mathcal{N} - \mathcal{N}(C_y)$, então $\alpha_{n,k}(C_y) = \max\{h : \min(q_y^{n,k-1}, q_y^{n,k}) \geq \gamma_y(C_{y,h})\}$;
- se $(n,i) \in C_y$, então $\alpha_{n,k}(C_y) = 1 + \max\{h : \min(q_y^{n,k-1}, q_y^{n,k}) - \min(q_y^{n,i-1}, q_y^{n,i}) \geq \gamma_y(C_{y,h}(n,i))\}$,

onde $\mathcal{N}(C_y) = \{n : (n,k) \in C_y\}$.

Proposição 6.3 A desigualdade de cobertura- y estendida e fortalecida:

$$\sum_{(n,k) \in C_y} x_{n,k} + \sum_{(n,k) \in E(C_y) - C_y} \alpha_{n,k}(C_y) x_{n,k} \leq |C_y| - 1 \quad (6.9)$$

é válida para \mathcal{P}_{cpl}^5 .

Prova: Sejam:

- $C_y^1 = \{(n,k) \in C_y : x_{n,k} = 1\}$, $C_y^0 = C_y - C_y^1$,
- $E_C^{1,h}(C_y) = \{(n,k) \in E(C_y) - C_y : n \in \mathcal{N}(C_y), x_{n,k} = 1, \alpha_{n,k}(C_y) = h\}$,
- $E_{\bar{C}}(C_y)^{1,h}(C_y) = \{(n,k) \in E(C_y) - C_y : n \notin \mathcal{N}(C_y), x_{n,k} = 1, \alpha_{n,k}(C_y) = h\}$,
- $E_C^1(C_y) = \bigcup_{h=1}^{|C_y|-1} E_C^{1,h}(C_y)$,
- $E_{\bar{C}}^1(C_y) = \bigcup_{h=1}^{|C_y|-1} E_{\bar{C}}^{1,h}(C_y)$,

- $\delta(n) = (n, k) \in C_y$ tal que $(n, j) \in E_C(C_y)$,
- $\Phi_1 = \{(n, k) \in C_y^0 : (n, j) \in E_C^1(C_y)\}$,
- $\bar{\Phi}_1 \subseteq \{(n, k) \in C_y^0 - \Phi_1\}$ é o subconjunto de máxima cardinalidade tal que $|\bar{\Phi}_1| \leq \sum_{(n,k) \in E_C^1(C_y)} (\alpha_{n,k}(C_y) - 1)$,
- $\Phi_2 \subseteq \{(n, k) \in C_y^0 - (\Phi_1 \cup \bar{\Phi}_1)\}$ é o subconjunto de máxima cardinalidade tal que $|\Phi_2| \leq \sum_{(n,k) \in E_C^1(C_y)} (\alpha_{n,k}(C_y))\}$.

Suponha que exista (x, λ) em \mathcal{P}_{cpl}^5 que viola (6.9). Então,

$$\begin{aligned}
 \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [q_y^{n,1} x_{n,k} - (q_y^{n,k} - q_y^{n,1}) \lambda_{n,k}] &\geq \\
 \sum_{(n,k) \in \Omega} \min\{q_y^{n,k-1}, q_y^{n,k}\} x_{n,k} &\geq \\
 \gamma_y(C_y^1) + \gamma_y(E_C^1(C_y)) + \gamma_y(E_C^1(C_y)) &\geq \\
 \gamma_y(C_y^1) + \sum_{h=1}^{|C_y|-1} \sum_{(n,k) \in E_C^{1,h}(C_y)} \gamma_y(C_{y,h}(\delta(n))) + \sum_{(n,k) \in E_C^{1,h}(C_y)} \gamma_y(C_{y,h}) &\geq \\
 \gamma_y(C_y^1) + \gamma_y(\Phi_1) + \gamma_y(\bar{\Phi}_1) + \gamma_y(\bar{\Phi}_2) = \gamma(C_y) &> q_y^{max}
 \end{aligned}$$

o que contradiz (6.5b) e, portanto, refuta a hipótese de que (x, λ) é factível. ■

Calcularemos agora os fatores $\alpha_{n,k}(C_i)$ do exemplo, com $E(C_i) - C_i = \{(2,5), (3,4), (3,5)\}$ e $C_i = \{(1,5), (2,4)\}$:

- $\alpha_{2,5}(C_i) = 1 + \max\{h : \min(q_i^{2,4}, q_y^{2,5}) - \min(q_i^{2,3}, q_y^{2,4}) \geq \gamma_i(C_{i,h}(2,4))\} = 1 + \max\{h : 6 - 4 \geq 7\} = 1 + 0 = 1$;
- $\alpha_{3,4}(C_i) = \max\{h : \min(q_i^{3,3}, q_i^{3,4}) \geq \gamma_i(C_{i,h})\} = \max\{h : 8 \geq \gamma_i(C_{i,h})\} = 1$;
- $\alpha_{3,5}(C_i) = \max\{h : \min(q_i^{3,4}, q_i^{3,5}) \geq \gamma_i(C_{i,h})\} = \max\{h : 9 \geq \gamma_i(C_{i,h})\} = 1$.

Sendo assim, a desigualdade de cobertura estendida e fortalecida para a cobertura C_i de nosso exemplo fica exatamente igual à equação (6.7).

Os desenvolvimentos aqui podem ser substancialmente simplificados quando os elementos de $\mathcal{Q}_y^n = \{q_y^{n,0}, \dots, q_y^{n,\kappa(n)}\}$ são estritamente crescentes, ou seja, se $q_y^{n,k} < q_y^{n,k+1}$ para $k = 0, \dots, \kappa(n) - 1$. Claramente, esta propriedade é satisfeita pelos elementos de \mathcal{Q}_i^n para todo $n \in \mathcal{N}$. Caso esta propriedade seja satisfeita para todos os \mathcal{Q}_y^n , as desigualdades de cobertura-y obtidas são semelhantes às aquelas desenvolvidas para $P(\emptyset)$ no Capítulo 5.

Da mesma maneira que para as variáveis x , limites inferiores para os fatores de *lifting* podem ser obtidos para as variáveis λ . Seja $\Omega_1 = \Omega \cup \{(n, 1) : n \in \mathcal{N}\}$ e $P(n, k) = \{(n, k-1), (n, k), (n, k+1)\} \cap \Omega_1$.

Definição 6.7 Dada uma cobertura C_y , os limites inferiores $\delta_{n,k}(C_y)$ para os fatores de *lifting* das variáveis λ para cada $(n, k) \in E_\lambda(C_y) - C_y$ são obtidos como segue:

- se $n \in \mathcal{N} - \mathcal{N}(C_y)$, então $\delta_{n,k}(C_y) = \max\{h : \min\{q_y^{n,j} : (n, j) \in P(n, k)\} \geq \gamma_y(C_{y,h})\}$;
- se $(n, i) \in C_y$, então $\delta_{n,k}(C_y) = 1 + \max\{h : \min\{q_y^{n,j} : (n, j) \in P(n, k)\} - \min(q_y^{n,i-1}, q_y^{n,i}) \geq \gamma_y(C_{y,h}(n, i))\}$.

Com os limites inferiores dos fatores de *lifting* das variáveis λ calculados para os elementos de $E_\lambda(C_y) - C_y$, uma desigualdade válida mais forte pode ser obtida.

Proposição 6.4 A desigualdade de cobertura- λ - y estendida e fortalecida de C_y :

$$\sum_{(n,k) \in E_\lambda(C_y) \cap C_y} \lambda_{n,k} + \sum_{(n,k) \in E_\lambda(C_y) - C_y} \delta_{n,k}(C_y) \lambda_{n,k} \leq |C_y| - 1 \quad (6.10)$$

é válida para \mathcal{P}_{cpl}^5 onde $\lambda_{n,1}$ é substituído pela Equação (6.4c).

6.3 Resultados Numéricos

Nesta seção são apresentados experimentos numéricos realizados com o intuito de demonstrar a eficiência dos planos de corte inseridos na formulação original.

Para instâncias do problema com 32 e 64 poços, introduzimos as desigualdades (6.1d)–(6.1g) com restrições na capacidade de separação e nos limites de armazenamento/processamento de óleo, gás e água. Os valores possíveis para q_i^{max} e q_p^{max} são *baixo*, *médio*, e *abundante*, mas os valores para q_o^{max} , q_g^{max} , e q_w^{max} são *baixo* e *médio* para que o número de experimentos seja relativamente baixo. Assim, o número total de experimentos para cada instância do problema foi $3^2 \times 2^3 \times 3 = 216$, pois realizamos um experimento sem cortes, um com cortes nas variáveis x , e outro incluindo os planos de cortes nas variáveis λ para cada situação. Os resultados destes experimentos são mostrados na Tabela 6.4 para o cenário com 32 poços e na Tabela 6.5 para o caso com 64 poços. As entradas das tabelas são as médias dos valores para q_o^{max} , q_g^{max} e q_w^{max} . Os planos de corte foram calculados utilizando a heurística mostrada na Seção 5.5 e aplicada para cada restrição do tipo mochila (6.1c)–(6.1g). Os resultados foram gerados utilizando o *GNU solver* [21] e um computador Pentium IV de 2.5GHz e 1GB de memória.

Tabela 6.4: Impacto das coberturas-y na velocidade de solução para instâncias com 32 poços.

q_i^{max}	q_p^{max}	Sem cortes		Com cortes-y em x		Com cortes-y em λ	
		Iterações	Nós	Iterações	Nós	Iterações	Nós
700	15.000	428.087	62.935	3.986	939	59	3
	20.000	428.087	62.935	3.986	939	59	3
	25.000	428.087	62.935	3.986	939	59	3
1.100	15.000	18.275	4.459	2.256	531	174	35
	20.000	281.357	61.587	10.385	2.819	81	1
	25.000	281.357	61.587	10.385	2.819	81	1
3.000	15.000	245	63	245	63	245	63
	20.000	3.958	1.041	932	189	108	3
	25.000	28.200	10.354	3.583	1.267	112	1
Média		210.850	36.432	4.416	1.167	109	13

Tabela 6.5: Impacto das coberturas-y na velocidade de solução para instâncias com 64 poços.

q_i^{max}	q_p^{max}	Sem cortes		Com cortes-y em x		Com cortes-y em λ	
		Iterações	Nós	Iterações	Nós	Iterações	Nós
1.100	40.000	*32.648.455	4.286.938	313.077	48.131	87	3
	50.000	*32.648.455	4.286.938	313.077	48.131	87	3
	70.000	*32.648.455	4.286.938	313.077	48.131	87	3
2.300	40.000	*25.968.268	3.598.636	690.753	161.517	169	1
	50.000	*23.741.320	3.372.521	690.753	161.517	169	1
	70.000	*23.741.320	3.372.521	690.753	161.517	169	1
3.500	40.000	279	99	279	99	279	99
	50.000	3.477	1.061	2.896	738	3.026	744
	70.000	169.911	79.362	24.143	6.833	9.107	1.949
Média		19.063.326	2.587.223	337.645	70.734	1.464	311

Das Tabelas 6.4 e 6.5, é evidente que os planos de corte propostos reduzem drasticamente o número de iterações (pivoteamentos de programação linear) e uso de memória dos algoritmos *branch-and-bound*. As instâncias marcadas com “*” não foram resolvidas até o fim devido à demanda exorbitante de recursos computacionais (o procedimento de *branch-and-bound* foi abortado após aproximadamente uma semana de tempo de execução). Estas instâncias são resolvidas facilmente após inserir na formulação as desigualdades de cobertura-y, especialmente se as coberturas-y estão nas variáveis λ .

6.4 Sumário

Neste capítulo, foi proposta uma formulação linear por partes para o problema de alocação de gás de injeção sob múltiplas restrições nas instalações de superfície, foi realizada uma análise do modelo e demonstrada sua aplicação prática. Os experimentos numéricos utilizando o *GNU solver* [21] mostraram que as desigualdades de cobertura- y podem tornar factíveis a otimização de instâncias médias e grandes utilizando *software* livre, o que de outra maneira seria excessivamente lento e consumiria recursos computacionais elevados. Os resultados desta seção foram publicados em [10].

Capítulo 7

Conclusões

Quase todos os setores da economia mundial dependem da indústria do petróleo, e com a demanda sempre crescente deste combustível fóssil, existe uma busca também crescente de melhor eficiência dos processos de produção, processamento e transporte de hidrocarbonetos. Técnicas de elevação artificial que aumentam o fator de recuperação de um reservatório de petróleo são amplamente utilizadas, e uma das técnicas mais utilizadas é a elevação artificial por injeção contínua de gás, ou *gas-lift*. Sendo assim, a otimização da alocação de gás de elevação para poços de petróleo desempenha um papel importante na eficiência de recuperação de um reservatório.

Os artigos encontrados na literatura confirmam esta tendência. Enquanto os estudos mais antigos estavam mais focados na determinação das curvas de desempenho dos poços, as investigações mais recentes abordam a operação ótima das instalações ao longo do tempo, lidando com mudanças nas condições de operação e as oscilações do mercado. Este trabalho contribui para o estado da arte dos sistemas de produção da indústria do petróleo, através da abordagem formal do problema da alocação ótima de gás de elevação, i.e., dado um grupo de poços de petróleo e um conjunto de restrições, decidir quais poços devem estar ativos e qual a taxa ótima de injeção de gás para cada poço de modo a obter um desempenho global ótimo. Foi mostrado que o problema de otimização não-linear inteira mista resultante ($P(\emptyset)$) é uma generalização do problema da mochila (portanto um problema NP-Difícil), e que captura todas as características dos problemas considerados em [15], [18] e [32].

Este documento foi organizado de modo a mostrar a evolução da complexidade dos problemas abordados. Primeiramente, tratamos de resolver o problema mais simples contido na literatura, onde há apenas a restrição da taxa disponível de injeção de gás. Foi obtido um algoritmo pseudo-polinomial baseado na técnica de Programação Dinâmica que resolve uma versão discretizada de P , onde *unidades padrão* de taxa de injeção de gás são alocadas entre os poços. Então, o problema foi incrementado de modo a capturar restrições de ativação e desativação de poços, que podem surgir por motivos técnicos ou estratégicos. Para esta versão mais complexa foi mostrado que não é possível desenvolver um algoritmo pseudo-polinomial, ou seja, o problema é NP-Difícil no sentido forte.

Apesar da natureza aproximada da solução recursiva, os resultados numéricos obtidos indicam que estes algoritmos são capazes de obter soluções quase-ótimas. Porém, a inclusão de novas restrições não é trivial.

Passamos então a utilizar outra abordagem, a linearização por partes das curvas de desempenho de *gas-lift*. Esta abordagem elimina as não-linearidades do problema, dando origem a um problema de otimização linear inteira mista, tornando mais flexível a introdução ou modificação de restrições, além de possibilitar o uso de algoritmos eficientes de Programação Inteira para a obtenção de soluções ótimas ou quase-ótimas. Esta estratégia foi sugerida pela primeira vez em [15], contudo, os autores não tiraram proveito da estrutura do problema para criar algoritmos específicos para o problema em questão, não considerando *WPCs* não-côncavas e tratando a ativação/desativação de poços de maneira *ad hoc*. Nosso trabalho abordou estas questões de maneira mais formal. Foi mostrado que o poliedro que descreve o espaço das soluções factíveis possui dimensão cheia quando são eliminadas as igualdades contidas na formulação, e que para o caso de curvas de desempenho côncavas, apenas uma variável inteira é necessária. Foram obtidas famílias de desigualdades válidas que são extensões das desigualdades de cobertura para o problema da mochila, e que ao serem introduzidas na formulação tornam mais rápida a busca de soluções ótimas por algoritmos de *branch-and-bound* e *branch-and-cut*. Resolvemos novamente o problema básico com a restrição da taxa disponível de injeção de gás, e também um problema mais completo, onde há múltiplas restrições que surgem devido a limitações diversas.

Ambas as abordagens foram testadas contra experimentos publicados na literatura, mostrando-se bastante eficientes. Os resultados deste trabalho representam excelentes ferramentas de suporte à decisão de baixo custo, pois podem ser implementadas em plataformas de *software* livre, além de poderem ser integradas a programas de simulação e controle.

7.1 Pesquisa Futura

A seguir, apontamos algumas lacunas deste trabalho que podem ser objeto de estudo em pesquisas futuras.

- 1. Formulação de $P^5(0)$ utilizando o modelo de Sherali:** as desigualdades válidas obtidas para $P^5(0)$ foram obtidas baseadas na formulação mais intuitiva, chamada de Modelo I (Seção 2.3.1). Porém, Padberg [33] mostra que este modelo não é o mais adequado. Os desenvolvimentos do Capítulo 6 podem ser reformulados de modo a seguir o modelo de Sherali [37].
- 2. Frações de óleo, gás e água variáveis:** uma simplificação da modelagem apresentada é a consideração de frações constantes de óleo, gás e água produzidos por um determinado poço n , γ_o^n , γ_g^n e γ_w^n respectivamente. Em situações reais, estas frações podem variar de acordo com a taxa

de injeção de gás q_i^n , pois a variação de q_i^n corresponde a uma variação na pressão no fundo do poço, o que por sua vez corresponde a uma variação nas frações de óleo e gás quando o petróleo é saturado. Diminuir a pressão no fundo do poço (ou seja, aumentar a taxa de injeção de gás) além de certos limites também pode elevar o contato óleo-água até a região dos canhoneados, aumentando a produção de água devido ao efeito de conificação. A introdução de frações variáveis em função de q_i^n pode dar origem a um problema de maximização não-convexo. O estudo de como as frações de óleo, gás e água produzidas por um poço variam em função da taxa de injeção de gás neste poço, bem como os seus efeitos na formulação do problema e nos algoritmos desenvolvidos para o cálculo de soluções ótimas será objeto de estudo.

- 3. Integração controle/otimização:** outra direção que deve ser seguida é a integração das técnicas de controle desenvolvidas em nosso grupo de trabalho [34] com as técnicas de otimização apresentadas no presente trabalho. Este trabalho de integração faz parte de uma proposta de dissertação de doutorado de um dos integrantes de nosso grupo de pesquisas, e deverá ser realizado em conjunto. O produto final do trabalho de mestrado citado deverá ser um ambiente gráfico de controle/otimização para o problema de elevação artificial via injeção contínua de gás.
- 4. Estudo de WPCs acopladas:** uma direção de pesquisa interessante é desenvolver formulações e algoritmos para o caso onde a taxa de produção de um poço k dependa das taxas de injeção de seus i poços vizinhos, ou seja, $q_p^k = q_p^k(q_i^1, q_i^2, \dots, q_i^N)$. Este modelo pode representar, por exemplo, cenários onde existam poços injetores de água.

Apêndice A

Ajuste de Curvas

Dado um conjunto de pontos de desempenho de *gas-lift* $Q^n = \{(q_i^{n,1}, q_p^{n,1}), (q_i^{n,2}, q_p^{n,2}), \dots, (q_i^{n,k}, q_p^{n,k})\}$, com $q_i^{n,j} < q_i^{n,j+1}$ para $j = 1, \dots, k-1$, aplicamos um método de ajuste de curvas e obtemos expressões analíticas $\tilde{q}_p^n(q_i^n)$ que representam de forma aproximada a função $q_p^n(q_i^n)$. Neste trabalho, foram utilizados três modelos de curvas:

- 1) Uma função polinomial da forma $\tilde{q}_p^n(q_i^n) = \alpha_0^n y_n + \alpha_1^n q_i^n + \alpha_2^n (q_i^n)^2 + \alpha_3^n (q_i^n)^3$, definida pelas constantes $\alpha_0^n, \alpha_1^n, \alpha_2^n$ e α_3^n . A vantagem deste modelo é que o problema de ajuste de curvas gerado é um problema de fácil solução. Sua desvantagem é a má qualidade de extrapolação para pontos fora do intervalo $[q_i^{n,1}, q_i^{n,k}]$;
- 2) Uma função exponencial da forma $\tilde{q}_p^n(q_i^n) = A_n(2 - e^{-B_n q_i^n}) - C_n e^{D_n q_i^n} + F_n(1 - y_n)$, definida pelas constantes A_n, B_n, C_n, D_n e $F_n = C_n - A_n$. Esta função exponencial é uma boa aproximação para o comportamento típico de uma *WPC*, fornece bons dados de extrapolação, e sua concavidade é garantida satisfazendo condições simples. Garantindo que as constantes A_n e C_n sejam positivas, garantimos também que a função $\tilde{q}_p^n(q_i^n)$ é uma função côncava, pois sob estas condições sua derivada segunda assume valores negativos para qualquer valor de q_i^n ;
- 3) Uma função logarítmica da forma $q_p^n(q_i^n) = c_1^n y_n + c_2^n q_i^n + c_3^n (q_i^n)^2 + c_4^n \ln(q_i^n + 1)$, definida pelas constantes c_1^n, c_2^n, c_3^n e c_4^n . Este modelo foi utilizado para comparar os resultados obtidos com o algoritmo PD e os obtidos com o método de SQP proposto em [1].

Para o primeiro caso (aproximação polinomial), o problema a ser resolvido é linear e portanto bastante simples, gerando o ótimo global:

$$\begin{aligned} \text{Min} \quad & f = \sum_{j=1}^k \left[\alpha_0^n y_n + \alpha_1^n q_i^{n,j} + \alpha_2^n (q_i^{n,j})^2 + \alpha_3^n (q_i^{n,j})^3 - q_p^{n,j} \right]^2 \\ \text{S. a:} \quad & 2\alpha_2^n + 6\alpha_3^n q_i^{n,j} \geq 0, \quad j = 1, \dots, k \end{aligned} \tag{A.1}$$

Encontrar os coeficientes A_n , B_n , C_n e D_n para o segundo método (aproximação exponencial) significa resolver o seguinte problema de mínimos quadrados para cada poço n :

$$\begin{aligned} \text{Min} \quad & f = \sum_{j=1}^k \left[A_n(2 - e^{-B_n q_i^{n,j}}) - C_n e^{D_n q_i^{n,j}} - q_p^{n,j} + F_n(1 - y_n) \right]^2 \\ \text{S. a:} \quad & \\ & A_n \geq 0 \\ & C_n \geq 0 \end{aligned} \tag{A.2}$$

Fornecemos soluções iniciais factíveis e utilizamos algoritmos de otimização não linear [27] para encontrar um ótimo local. As curvas encontradas passam por um processo de seleção gráfico e, para cada conjunto Q^n obtemos os coeficientes A_n , B_n , C_n e D_n . As curvas obtidas são côncavas para todo o intervalo $q_i^n \in [0, u_n]$.

O terceiro método de ajuste de curvas (aproximação logarítmica) consiste em resolver o seguinte problema para cada poço n :

$$\begin{aligned} \text{Min} \quad & f = \sum_{j=1}^k \left[c_1^n y_n + c_2^n q_i^{n,j} + c_3^n (q_i^{n,j})^2 + c_4^n \ln(q_i^{n,j} + 1) - q_p^{n,j} \right]^2 \\ \text{S. a:} \quad & \\ & 2c_3^n - \frac{c_4^n}{(1+q_i^{n,j})^2} \leq 0 \quad \text{para } j = 1, \dots, k \end{aligned} \tag{A.3}$$

A resposta desse problema são as constantes c_1^n , c_2^n , c_3^n e c_4^n . As curvas obtidas são côncavas na região¹ $q_i = [l_n, u_n]$.

¹Não é necessário garantir concavidade no intervalo $q_i^n \in [0, l_n]$ pois não foi calculada a relaxação linear para este caso.

Apêndice B

Desigualdades Válidas para $P(G)$

Seja $\mathcal{P}(G)$ o poliedro correspondente ao fecho convexo das soluções factíveis para $P(G)$ no qual q_p^n é substituído pela sua função de q_i^n , i.e., $\mathcal{X}(G) = \{(q_i, y) \in \mathbb{R}^N \times \mathbb{B}^N : \text{sujeito às restrições de (4.1)}\}$ e $\mathcal{P}(G) = \text{conv}(\mathcal{X}(G))$. Para os desenvolvimentos a seguir, a terminologia de grafos introduzida no Capítulo 4 necessita ser estendida. Seja $\lambda_n = \sum_{k \in S(n)} l_k$ a mínima taxa de injeção necessária para ativar o poço n e seus descendentes em G . Seja $W(C) = \cup_{n \in C} S(n)$ o conjunto de todos os poços que precisam ser ativados para que os poços em $C \subseteq \mathcal{N} = \{1, \dots, N\}$ sejam ativados.

Proposição B.1 *Se G é um grafo acíclico, $l_n < u_n < q_i^{max}$ para todo n , e $\lambda_n < q_i^{max}$ para todo $n \in \text{root}(G)$, então $\mathcal{P}(G) = \mathcal{X}(G)$ possui dimensão cheia, isto é, $\dim(\mathcal{P}(G)) = 2N$.*

Prova: Seja $L = \langle v_1, \dots, v_N \rangle$ uma ordem topológica para G . A seguir mostramos que vetores unitários para q_i^n e y_n podem ser obtidos sistematicamente, a partir de elementos de $\mathcal{X}(G)$, na ordem reversa de L . Mais formalmente, o procedimento encontra vetores unitários para as variáveis associadas ao poço v_N , e então encontra um vetor unitário para o poço v_{N-1} , e assim por diante até chegar ao poço v_1 . Os passos do procedimento são descritos abaixo.

- **Passo 1.** Seja $U = \emptyset$ o conjunto dos índices dos poços para os quais vetores unitários de suas variáveis tenham sido obtidos. Faça $k = N$.
- **Passo 2.** Faça $n = v_k$ e seja $\tilde{z} = (\tilde{q}_i, \tilde{y}) \in \mathcal{X}(G)$ a solução obtida fazendo $\tilde{y}_j = 1$ e $\tilde{q}_i^j = l_j$ para cada $j \in S(n)$, além de $\tilde{y}_j = 0$ e $\tilde{q}_i^j = 0$ para cada $j \notin S(n)$. Seja $\hat{z} = (\hat{q}_i, \hat{y}) \in \mathcal{X}(G)$ a solução obtida fazendo $\hat{y}_j = 1$ e $\hat{q}_i^j = l_j$ para cada $j \in S^+(n)$, além de $\hat{y}_n = 1$ e $\hat{q}_i^n = \min\{u_n, q_i^{max} - (\lambda_n - l_n)\}$, e $\tilde{y}_j = 0$ e $\tilde{q}_i^j = 0$ para cada $j \notin S(n)$. Note que as soluções \tilde{z} e \hat{z} são idênticas em todas as suas entradas, com exceção do poço n , mais precisamente $\tilde{q}_i^n < \hat{q}_i^n$ pois $\lambda_n < q_i^{max}$. Por hipótese de indução, podemos assumir que os vetores unitários foram obtidos para todas as entradas diferentes de zero de \tilde{z} a menos daquelas correspondentes ao poço n , pois estes poços são precedidos pelo poço n na ordem topológica. Por combinação linear destes vetores unitários e

das soluções \tilde{z} e \hat{z} , podemos obter vetores unitários para as variáveis q_i^n e y_n . Portanto, U pode ser estendido para englobar o poço n , $U \leftarrow U \cup \{n\}$. Se $k = 1$ então pare, senão decmente k por 1 e repita a partir do Passo 2.

Neste ponto, $U = \{1, \dots, N\}$, o que significa que vetores unitários foram obtidos a partir de combinação linear dos elementos de $X(G)$ para todas as variáveis. Como a origem pertence a $X(G)$, concluímos que $\mathcal{P}(G)$ contém $2N + 1$ elementos afim independentes, implicando em $\dim(\mathcal{P}(G)) = 2N$. ■

A partir deste ponto, assumimos que as condições da Definição B.1 são satisfeitas.

Definição B.1 Um conjunto $C \subseteq \{1, \dots, N\}$ é uma cobertura para $X(G)$ se:

- $S(n) \cap C = \{n\}$ para cada $n \in C$;
- $\sum_{n \in C} \lambda_n > q_i^{\max}$;
- $\sum_{n \in C} \lambda_n - l_k < q_i^{\max}$ para cada $k \in C$; e

Definição B.2 Seja $\mathcal{P}_C(G) = \{(q_i, y) \in \mathcal{P}(G) : y_n = q_i^n = 0\}$ para todo $n \in \mathcal{N} - W(C)$ o poliedro obtido pela projeção de $\mathcal{P}(G)$ sobre o espaço das variáveis y_n e q_i^n cujos índices pertencem ao conjunto $W(C)$, onde C é uma cobertura.

Corolário B.1 Se C é uma cobertura, então $\dim(\mathcal{P}_C(G)) = 2|W(C)|$.

Proposição B.2 Se C é uma cobertura, então a desigualdade de cobertura

$$\sum_{n \in C} y_n \leq |C| - 1 \quad (\text{B.1})$$

é válida para $\mathcal{P}_C(G)$ e, além disso, a face $F_C = \{(q_i, y) \in \mathcal{P}_C(G) : \sum_{n \in C} y_n = |C| - 1\}$ induzida pela desigualdade de cobertura possui dimensão $\dim(F_C) = 2|W(C)| - 1$.

Prova:. (Validade) Se $\sum_{n \in C} y_n = |C|$ para alguma solução $(q_i, y) \in X(G) \cap \mathcal{P}_C(G)$, então $\sum_{n \in W(C)} q_i^n \geq \sum_{n \in C} \lambda_n > q_i^{\max}$, mas isso significa que a solução é infcatível, contradizendo a hipótese.

(Dimensionalidade) Seja:

$$\sum_{n \in W(C)} (\pi_n y_n + \mu_n q_i^n) \leq \pi_o \quad (\text{B.2})$$

uma desigualdade válida geral para $\mathcal{P}_C(G)$ induzindo a face máxima $F_\pi = \{(q_i, y) \in \mathcal{P}_C(G) : \sum_{n \in W(C)} (\pi_n y_n + \mu_n q_i^n) = \pi_o\}$ tal que $F_C \subseteq F_\pi$. A seguir provamos que F_C é máxima mostrando que a desigualdade (B.1)

(que induz F_C) difere da desigualdade (B.2) (que induz F_π) por uma constante multiplicativa positiva. A demonstração consiste em mostrar que $\pi_n = 1$ para todo $n \in C$, $\pi_n = 0$ para todo $n \in W(C) - C$, $\mu_n = 0$ para todo $n \in W(C)$ e $\pi_0 = |C| - 1$.

Primeiro, mostramos que $\mu_n = 0$ para todo $n \in C$. Seja $\tilde{z} = (\tilde{q}_i, \tilde{y})$ obtido fazendo $\tilde{y}_t = 1$ e $\tilde{q}_i^t = l_t$ para todo $t \in W(C - \{k\})$, onde $k \in C - \{n\}$, e $\tilde{y}_t = 0$ e $\tilde{q}_i^t = 0$ para todo $t \in S(k)$. Claramente, $\tilde{z} \in F_C$ e $\sum_{t \in W(C)} \tilde{q}_i^t = \sum_{t \in C - \{k\}} \lambda_t < q_i^{max}$. Similarmente, seja \hat{z} idêntico a \tilde{z} a menos de $\hat{q}_i^n = \min\{u_n, q_i^{max} - \sum_{t \in W(C) - \{n\}} \tilde{q}_i^t\}$. Para \tilde{z} e \hat{z} pertencerem a F_π , (B.2) deve ser satisfeita na igualdade. Subtraindo (B.2) calculada em \tilde{z} de (B.2) calculada em \hat{z} , e levando em conta o fato que $\hat{q}_i^n > \tilde{q}_i^n$, pode-se concluir que $\mu_n = 0$. Repetindo os passos acima para os elementos restantes da cobertura, verifica-se que $\mu_n = 0$ para todo $n \in C$.

Segundo, mostramos que $\mu_n = 0$ para $n \in W(C) - C$, onde $n \in S^+(l)$ para algum $l \in C$. Seja $\tilde{z} = (\tilde{q}_i, \tilde{y})$ obtido fazendo $\tilde{y}_t = 1$ e $\tilde{q}_i^t = l_t$ para todo $t \in W(C - \{k\})$, onde $k \in C - \{l\}$, e $\tilde{y}_t = 0$ e $\tilde{q}_i^t = 0$ para todo $t \in S(k)$. Segue que \tilde{z} pertence a F_C e $\sum_{t \in W(C)} \tilde{q}_i^t = \sum_{t \in C - \{k\}} \lambda_t < q_i^{max}$. Seja $\hat{z} \in F_C$ idêntico a \tilde{z} , exceto por \hat{q}_i^n que assume valor $\min\{u_n, q_i^{max} - \sum_{t \in W(C) - \{n\}} \tilde{q}_i^t\}$. Note que (B.2) deve ser satisfeita na igualdade para \tilde{z} e \hat{z} pois $\tilde{z}, \hat{z} \in F_\pi$. Subtraindo (B.2) calculada em \tilde{z} de (B.2) calculada em \hat{z} , e usando o fato de que $\hat{q}_i^n > \tilde{q}_i^n$, pode-se deduzir que $\mu_n = 0$. Repetindo os passos acima para os poços restantes, pode-se concluir que $\mu_n = 0$ para todo $n \in W(C) - C$. Neste ponto, descobrimos que a desigualdade que induz F_π é da forma $\sum_{n \in W(C)} \pi_n y_n \leq \pi_0$.

Terceiro, mostramos que $\pi_n = 0$ para todo $n \in W(C) - C$. Isto, é mostrado por indução e explorando a estrutura imposta pelas restrições de precedência. Seja $L = \langle v_1, \dots, v_{|W(C)-C|} \rangle$ a permutação dos elementos de $W(C) - C$ tal que v_j precede v_k em uma ordem topológica de G se e somente se $j > k$. Em outras palavras, os elementos de $W(C) - C$ são arranjados em L na ordem reversa da ordem topológica. Pode ser mostrado por indução em $l \in \{1, \dots, |W(C) - C|\}$ que $\pi_{v_l} = 0$ (*base de indução*), $\pi_{v_2} = 0$ (*passo de indução*), e assim por diante até que a afirmação seja demonstrada. Omitimos a base de indução e mostramos apenas o passo de indução. Para algum $l \geq 2$, seja $n = v_l$ onde $n \in S(k)$ para algum $k \in C$. Seja $\tilde{z} = (\tilde{q}_i, \tilde{y})$ obtido fazendo $\tilde{y}_t = 1$ e $\tilde{q}_i^t = l_t$ para todo $t \in W(C - \{k\}) \cup S(n)$, enquanto $\tilde{y}_t = 0$ e $\tilde{q}_i^t = 0$ para todo $t \in S(k) - S(n)$. Como $\tilde{y}_k = 0$ e $\tilde{y}_t = 1$ para todo $t \in S(n)$ segue que $\tilde{z} \in F_C$. Além disso, $\sum_{t \in W(C)} \pi_t \tilde{y}_t = \sum_{t \in W(C) - S(k)} \pi_t \tilde{y}_t + \sum_{t \in S(n)} \pi_t \tilde{y}_t = \pi_0$ para que \tilde{z} pertença a F_π . Por indução, segue que $\pi_t = 0$ para todo $t \in S(n) - \{n\}$. Assim:

$$\sum_{t \in W(C) - S(k)} \pi_t \tilde{y}_t + \pi_n \tilde{y}_n = \pi_0 \quad (B.3)$$

Agora, seja $\hat{z} \in F_C$ igual a \tilde{z} , com exceção de $\hat{y}_t = \hat{q}_i^t = 0$ para todo $t \in S(n)$. A equação (B.3) calculada em \hat{z} deve ser satisfeita para que \hat{z} pertença a F_π . Subtraindo (B.3) calculada em \hat{z} de (B.3) calculada em \tilde{z} , segue que $\pi_n = 0$. Repetindo os passos acima para $l+1, l+2, \dots, |W(C) - C|$, podemos deduzir que $\pi_n = 0$ para todo $n \in W(C) - C$, o que nos leva a concluir que a desigualdade (B.2) que induz F_π

é da forma:

$$\sum_{n \in C} \pi_n y_n \leq \pi_0 \quad (\text{B.4})$$

Quarto, é fácil verificar que $\pi_n = \pi_k = \pi$ para todo $n, k \in C, n \neq k$. Apenas produza dois elementos de F_C , um ativando o poço n mas não k , digamos \tilde{z} , e outro que ativa k mas não n , digamos \hat{z} . Para ter $\tilde{z}, \hat{z} \in F_\pi$, a desigualdade (B.4) deve ser satisfeita na igualdade se calculada em \tilde{z} e \hat{z} . Subtraindo (B.4) calculada em \tilde{z} de (B.4) calculada em \hat{z} , concluímos que $\pi_n = \pi_k$. Consequentemente, $\pi_0 = (|C| - 1)\pi$.

Fazendo $\pi_0 = |C| - 1$ e $\pi_n = 1$ para todo $n \in C$, deduzimos que a desigualdade (B.1) induz uma face máxima de $\mathcal{P}_C(G)$. Portanto $\dim(F_C) = 2|W(C)| - 1$. ■

De acordo com a Proposição B.2, a desigualdade induzida por uma cobertura C seria uma faceta de $\mathcal{P}(G)$ se $W(C) = \mathcal{N}$. Aplicando o princípio de *lifting*, é possível estender uma desigualdade de cobertura para $\mathcal{P}_U(G) = \mathcal{P}(G) \cap \{(q_i, y) \in \mathbb{R}^N \times \mathbb{B}^N : y_n = 0 \text{ para todo } n \in \mathcal{N} - U\}$, onde $U = W(C)$, e obter uma desigualdade válida para $\mathcal{P}(G)$. Como no procedimento de *lifting* mostrado para obter desigualdades válidas para \mathcal{P} (Seção 3.5), as variáveis associadas aos elementos de $\mathcal{N} - U$ seriam submetidas ao procedimento de *lifting* uma de cada vez. Seja $s = \langle j_1, \dots, j_L \rangle$ uma sequência ordenada dos elementos de $\mathcal{N} - U$. Para fazer o *lifting* do primeiro elemento desta sequência, j_1 , e produzir a desigualdade $\sum_{n \in C} y_n + \alpha_{j_1} y_{j_1} \leq |C| - 1$ que é válida para $\mathcal{P}_{U \cup \{j_1\}}(G)$, o procedimento calcularia $\alpha_{j_1} = |C| - 1 - \max\{\sum_{n \in C} y_n : (q_i, y) \in \mathcal{P}_{U \cup \{j_1\}}(G), y_{j_1} = 1\}$ resolvendo um problema relacionado a $P(G)$. A desigualdade resultante induz uma face $F_{U \cup \{j_1\}}$ de $\mathcal{P}_{U \cup \{j_1\}}(G)$ tal que $\dim(F_{U \cup \{j_1\}}) \geq \dim(F_U) + 1$. Se existe uma solução ótima para o problema de calcular α_{j_1} tal que $q_i^{j_1} > l_{j_1}$, então não seria difícil mostrar que a face induzida possui dimensão $\dim(F_{U \cup \{j_1\}}) = \dim(F_U) + 2$. Repetindo este procedimento para os elementos restantes de s , pode-se produzir uma desigualdade mais forte, que possivelmente induz uma faceta para $\mathcal{P}(G)$ chamada de *desigualdade de cobertura com lifting*:

$$\sum_{n \in C} y_n + \sum_{k=1}^L \alpha_{j_k} y_{j_k} \leq |C| - 1 \quad (\text{B.5})$$

A desigualdade (B.5) depende tanto da cobertura C como da ordem na qual os elementos de $\mathcal{N} - U$ sofrem o *lifting*. Como o cálculo dos fatores de *lifting* é difícil, pode-se tentar obter limites inferiores β_n para os fatores α_n . No que segue descrevemos uma maneira simples para calcular limites inferiores. Um subconjunto $E(C) \subseteq \mathcal{N}$ é uma cobertura estendida de C se:

1. $C \subset E(C)$;
2. para todos os pares distintos $n, m \in E(C)$, $S(n) \cap S(m) = \emptyset$; e
3. para cada $n \in E(C) - C$, $\lambda_n \geq \max\{\lambda_j : j \in C\}$.

Então, pode ser mostrado que a *desigualdade de cobertura estendida*:

$$\sum_{n \in E(C)} y_n \leq |C| - 1 \quad (\text{B.6})$$

é válida para $\mathcal{P}(G)$. Note que em (B.6) $\beta_n = 1$ para todo $n \in E(C) - C$ enquanto $\beta_n = 0$ para todo $n \in \mathcal{N} - E(C)$. Como nos desenvolvimentos anteriores, podemos obter limites inferiores de dimensões maiores para os fatores de *lifting*. Para $k \in \{0, \dots, |C|\}$, seja $C(k) \subseteq C$ tal que $|C(k)| = k$ e $\min\{\lambda_n : n \in C(k)\} \geq \max\{\lambda_n : n \in C - C(k)\}$ se $0 < k < |C|$, ou seja, $C(k)$ é o subconjunto de C que contém os poços que possuem as maiores taxas mínimas de injeção para serem ativados. Seja também $q(C') = \sum_{n \in C'} \lambda_n$ para qualquer $C' \subseteq C$. Para $n \in E(C) - C$, podemos definir um limite inferior β_n como:

$$\begin{aligned} \beta_n &= \text{Max}_{k \in \{1, \dots, |C|\}} \\ \text{S. a : } &\lambda_n \geq q(C(k)) \end{aligned}$$

Substituindo os fatores de *lifting* α_n em (B.5) por β_n para $n \in E(C) - C$ obtemos a *desigualdade de cobertura estendida e fortalecida*:

$$\sum_{n \in C} y_n + \sum_{n \in E(C) - C} \beta_n y_n \leq |C| - 1 \quad (\text{B.7})$$

que é válida para $\mathcal{P}(G)$.

Referências Bibliográficas

- [1] G. A. Alarcón, C. F. Torres, and L. E. Gómez. Global optimization of gas allocation to a group of wells in artificial lift using nonlinear constrained programming. *ASME Journal of Energy Resources Technology*, 124(4):146–164, 2002.
- [2] M. Assayag, J. Formigli, and M. V. Coelho. Águas ultraprofundas no Brasil: Os próximos passos para alcançar 3000 metros. *Revista T&N Petróleo*, Outubro 2000.
- [3] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.
- [4] E. Balas and E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, (34):119–148, 1978.
- [5] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.
- [6] S. Buitrago, E. Rodríguez, and D. Espin. Global optimization techniques in gas allocation for continuous flow gas lift systems. In *Gas Technology Conference*, Calgary, Canada, April 1996. Paper SPE 35616.
- [7] E. Camponogara and A. M. de Conto. Gas-lift allocation under precedence constraints: piecewise-linear formulation and k-covers. In *44th IEEE Conference on Decision and Control*, 2005.
- [8] E. Camponogara and P. Nakashima. Optimizing gas-lift production of oil wells: piecewise linear formulation and computational analysis. *IIE Transactions*, 38(2):173–182, 2006.
- [9] E. Camponogara and P. H. R. Nakashima. Solving a gas-lift optimization problem by dynamic programming. *European Journal of Operational Research*, 174:1220–1246, 2006.
- [10] E. Camponogara and P. H. R. Nakashima. Optimal allocation of lift-gas rates under multiple facility constraints: A mixed integer linear programming approach. *ASME Journal of Energy Resources Technology*, 128:280–289, 2006.
- [11] E. Camponogara and P. H. R. Nakashima. Applying dynamic programming to a gas-lift optimization problem. In *II Congresso de Pesquisa e Desenvolvimento em Petróleo e Gás*, Rio de Janeiro, RJ, 2003.

- [12] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [13] A. M. de Conto. Alocação ótima de gás de injeção em poços de petróleo: Restrições de precedência, linearização por partes e algoritmos. Master's thesis, Universidade Federal de Santa Catarina, 2006.
- [14] M. J. Economides, A. D. Hill, and C. Ehlig-Economides. *Petroleum Production Systems*. Prentice Hall, 1993.
- [15] W. Y. Fang and K. K. Lo. A generalized well-management scheme for reservoir simulation. *SPE Reservoir Engineering*, 11(2):116–120, May 1996. Paper SPE 29124.
- [16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [17] O. H. Ibarra and C. E. Kim. Scheduling for maximum profit. *Relatório Técnico 75-2, Universidade de Minnesota*, 1975.
- [18] E. P. Kanu, J. Mach, and K. E. Brown. Economic approach to oil production and gas allocation in continuous gas lift. *Journal of Petroleum Technology*, pages 1887–1892, October 1981. Paper SPE 9084.
- [19] K. K. Lo. Optimum lift-gas allocations under multiple production constraints. 1992. Paper SPE 26017.
- [20] E. M. Macambira. Uma abordagem de programação linear inteira para o problema de clique máxima com peso nas arestas. Master's thesis, Departamento de Ciência da Computação, IMECC, Universidade Estadual de Campinas, 1997.
- [21] A. Makhorin. *GNU Linear Programming Kit: Reference Manual*. Department of Applied Informatics, Moscow Aviation Institute, Moscow, Russia, 2003.
- [22] S. Martelo and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.
- [23] E. R. Martinez, W. J. Moreno, J. A. Moreno, and R. Maggiolo. Application of genetic algorithm on the distribution of gas lift injection. In *3rd Latin American/Caribbean Petroleum Engineering Conference*, Buenos Aires, Argentina, April 1994. Paper SPE 24281.
- [24] C. J. N. McKie, E. A. Rojas, N. M. Quintero, J. R. C. Fonseca, and N. J. Perozo. Economic benefits from automated optimization of high pressure gas usage in an oil production system. In *SPE Production and Operations Symposium*, Oklahoma, USA, 2001. Paper SPE 67187.
- [25] K. Mehlhorn and S. Naher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

- [26] H. Mukherjee and K. E. Brown. Improve your gas lift design. In *SPE International Meeting on Petroleum Engineering*, Beijing, China, March 1986. Paper SPE 14053.
- [27] B. H. Murtagh and M. A. Saunders. *MINOS 5.5 User's Guide*. Systems and Optimization Laboratory, Department of Operations Research, Stanford University, 1998.
- [28] P. H. R. Nakashima and E. Camponogara. Otimização da alocação de gás de injeção para um conjunto de poços operando via gas-lift. In *Congresso Brasileiro de Automática*, Gramado, RS, 2004.
- [29] P. H. R. Nakashima and E. Camponogara. Optimization of lift-gas allocation with uncertainties using dynamic programing. *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 36(2):407–414, 2006.
- [30] P. H. R. Nakashima and E. Camponogara. Alocação Ótima de taxas de injeção de gas-lift utilizando programação linear inteira mista. *Revista Petro & Química*, 263:150–160, Agosto 2004.
- [31] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [32] N. Nishikiori, R. A. Redner, D. R. Doty, and Z. Schmidt. An improved method for gas lift allocation optimization. In *SPE 64th Annual Technical Conference and Exhibition*, San Antonio, USA, October 1989. Paper SPE 19711.
- [33] M. Padberg. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, 27:1–5, 2000.
- [34] A. Plucenio. Automação da produção de poços de petróleo operando com elevação artificial por injeção contínua de gás. Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, 2003.
- [35] J. D. Redden, T. A. G. Sherman, and J. R. Blann. Optimizing gas-lift systems. In *49th Annual Fall Meeting of Petroleum Engineers of AIME*, Houston, USA, October 1974. Paper SPE 5150.
- [36] *GOAL User Guide*. Schlumberger Abingdon Technology Center, Oxon, United Kingdom, 2003.
- [37] H.D. Sherali. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Operations Research Letters*, 28:155–160, 2001.
- [38] S. S. Skiena. *The Algorithm Design Manual*. Springer Velag, 1998.
- [39] J. E. Thomas. *Fundamentos de Engenharia de Petróleo*. Editora Interciência, 2001.
- [40] P. Wang, M. Litvak, and K. Aziz. Optimization of production from mature fields. In *17th World Petroleum Congress*, Rio de Janeiro, Brasil, 2002.

-
- [41] H. P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, 1999.
- [42] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.